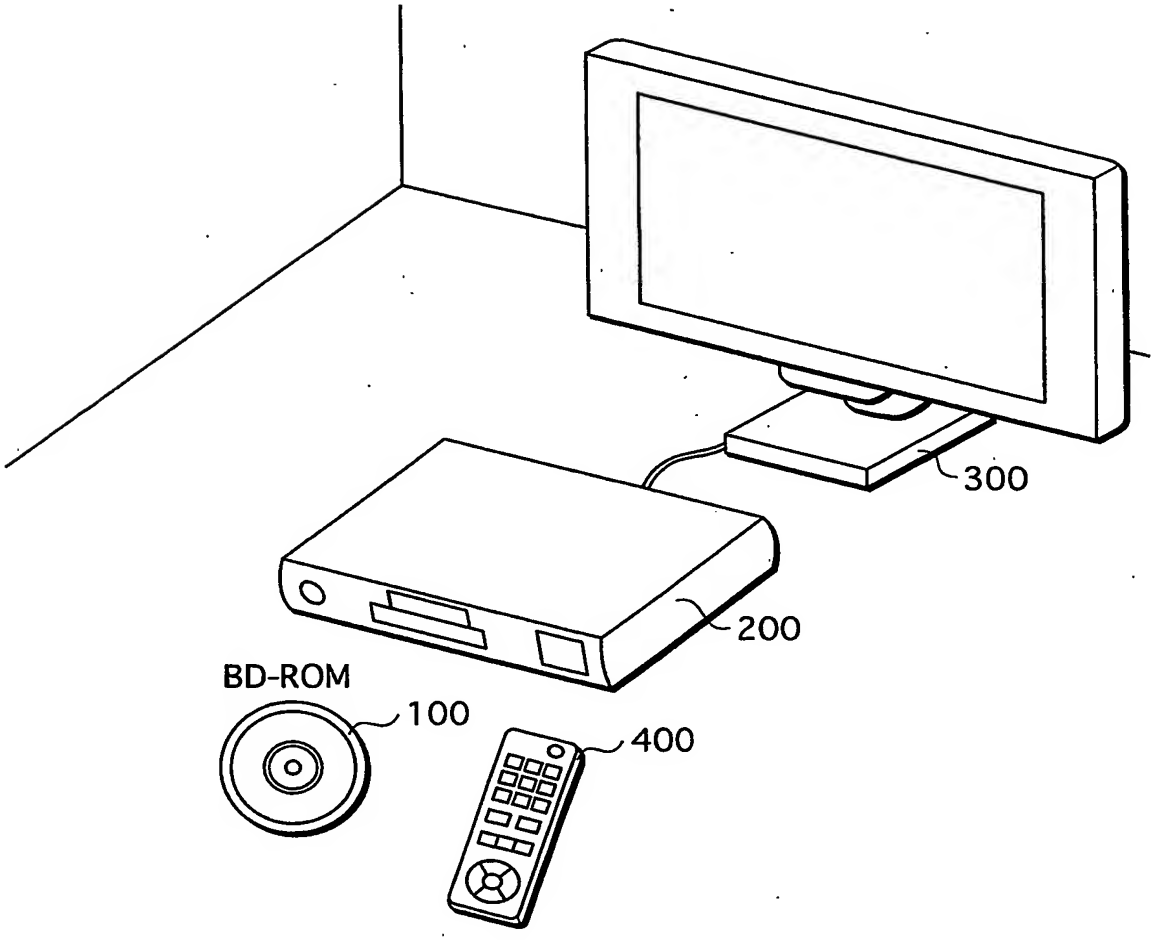
FIG. 1



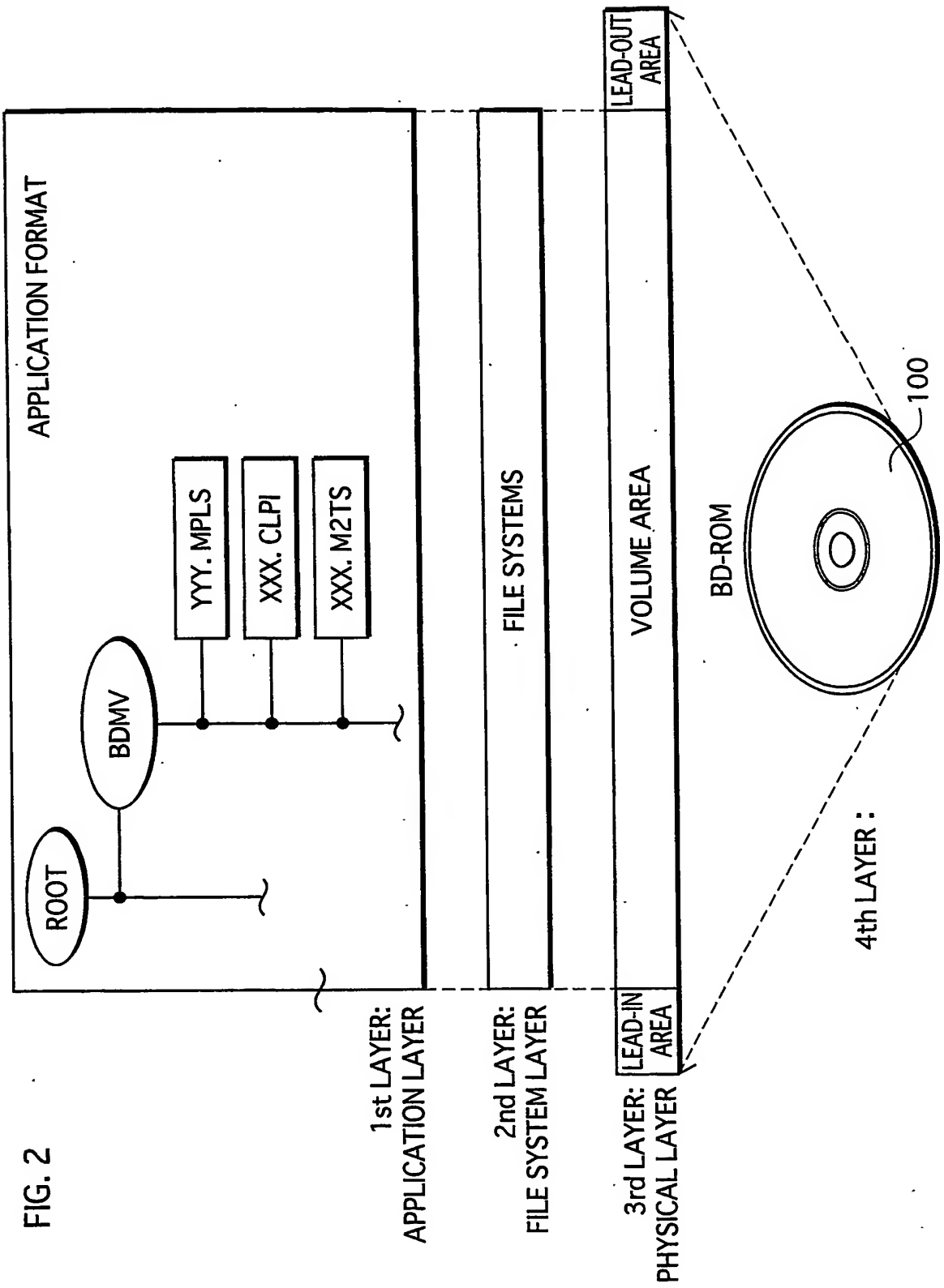
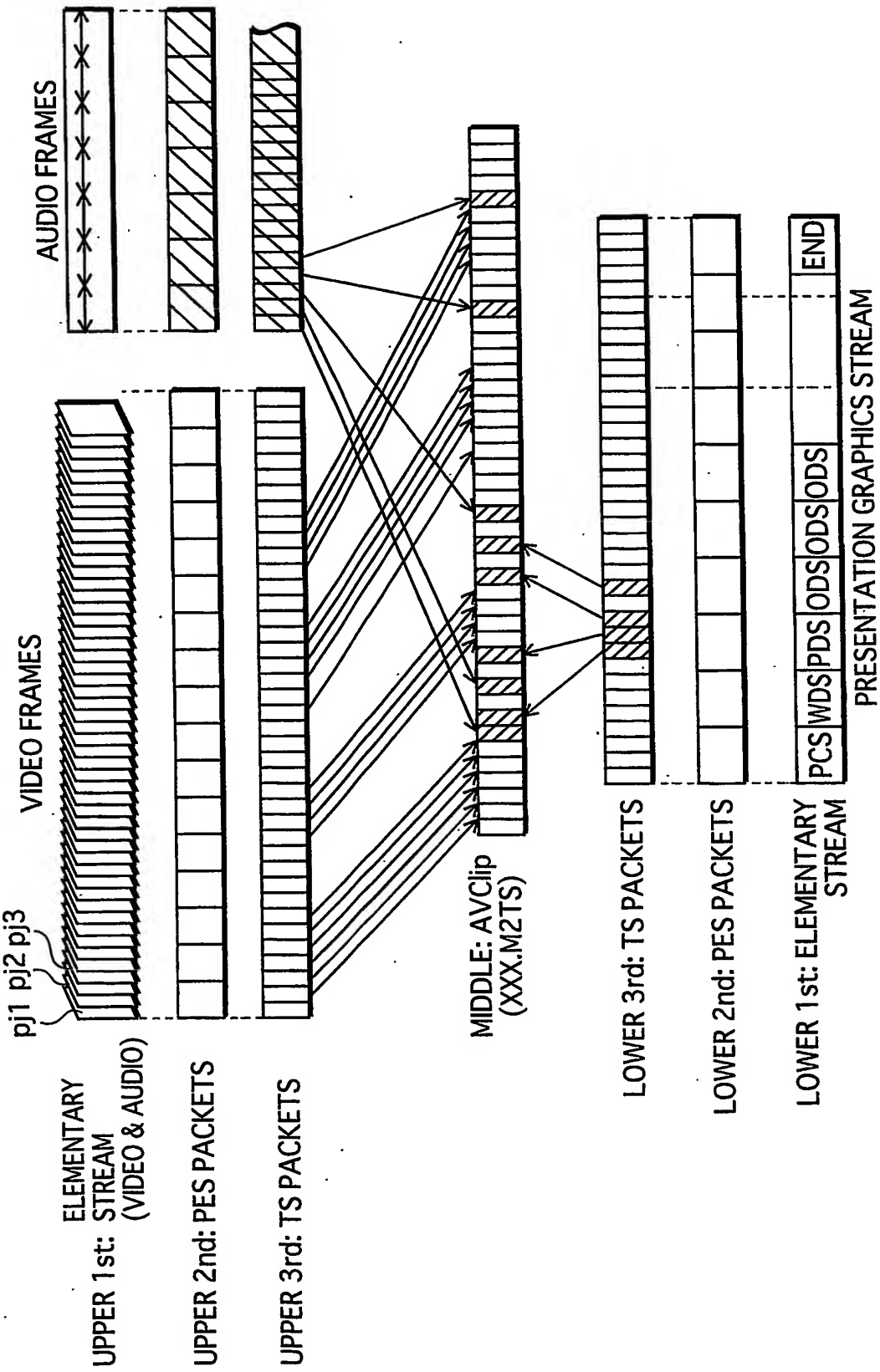

APPLICATION FORMAT
ROOT
BDMV
YYY. MPLS
XXX. CLPI
XXX. M2TS
FILE SYSTEMS
VOLUME AREA
LEAD-OUT AREA
LEAD-IN AREA
BD-ROM
100
4th LAYER :
1st LAYER: APPLICATION LAYER
2nd LAYER: FILE SYSTEM LAYER
3rd LAYER: PHYSICAL LAYER

FIG. 2

FIG. 3


AUDIO FRAMES
VIDEO FRAMES
pj1 pj2 pj3
UPPER 1st: ELEMENTARY STREAM (VIDEO & AUDIO)
UPPER 2nd: PES PACKETS
UPPER 3rd: TS PACKETS
MIDDLE: AVClip (XXX.M2TS)
LOWER 3rd: TS PACKETS
LOWER 2nd: PES PACKETS
LOWER 1st: ELEMENTARY STREAM
PCS
WDS
PDS
ODS
ODS
ODS
END
PRESENTATION GRAPHICS STREAM

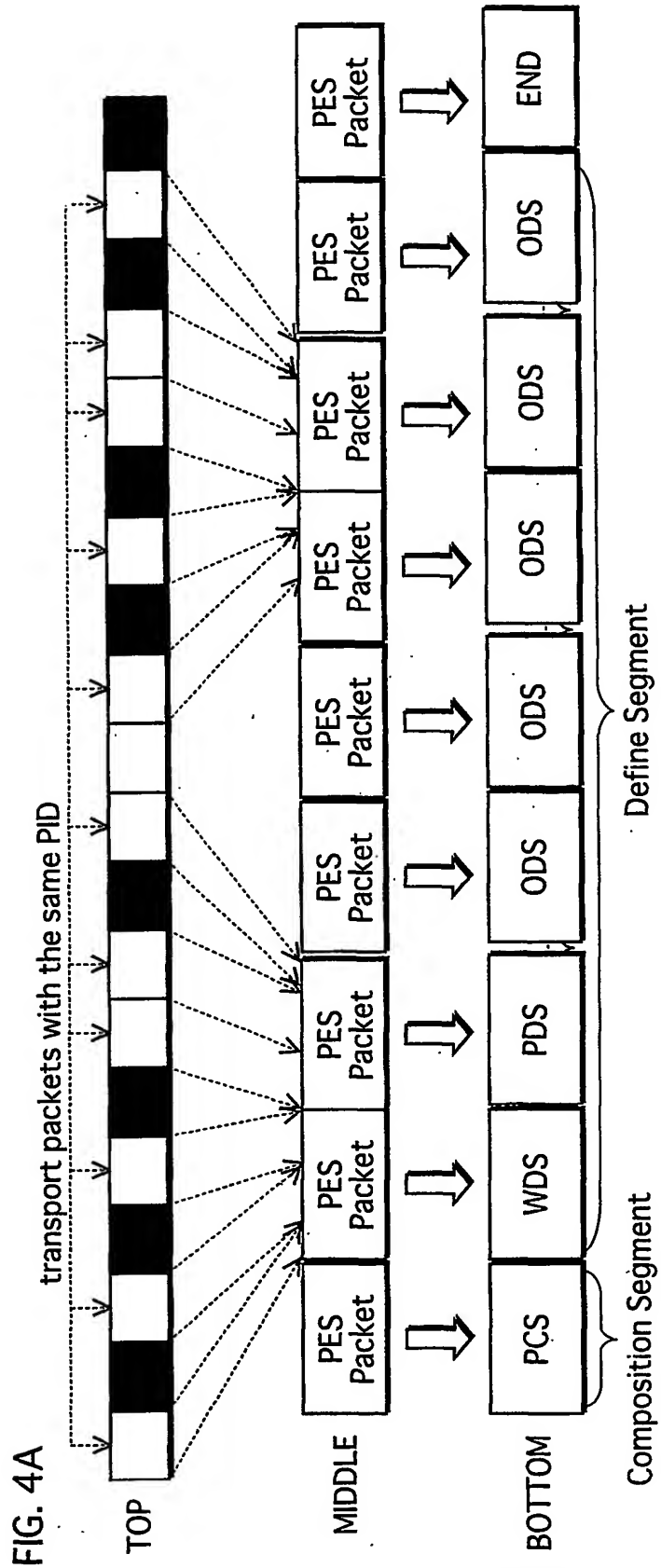

transport packets with the same PID
TOP
MIDDLE
BOTTOM
PES Packet
PCS
WDS
PDS
ODS
END
Composition Segment
Define Segment

FIG. 4A

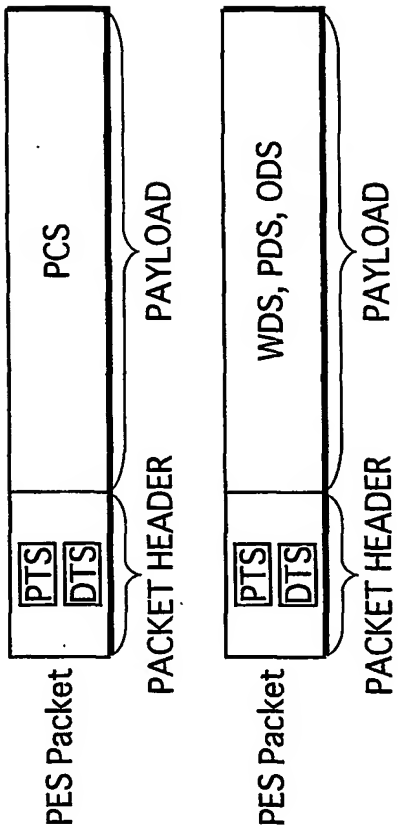
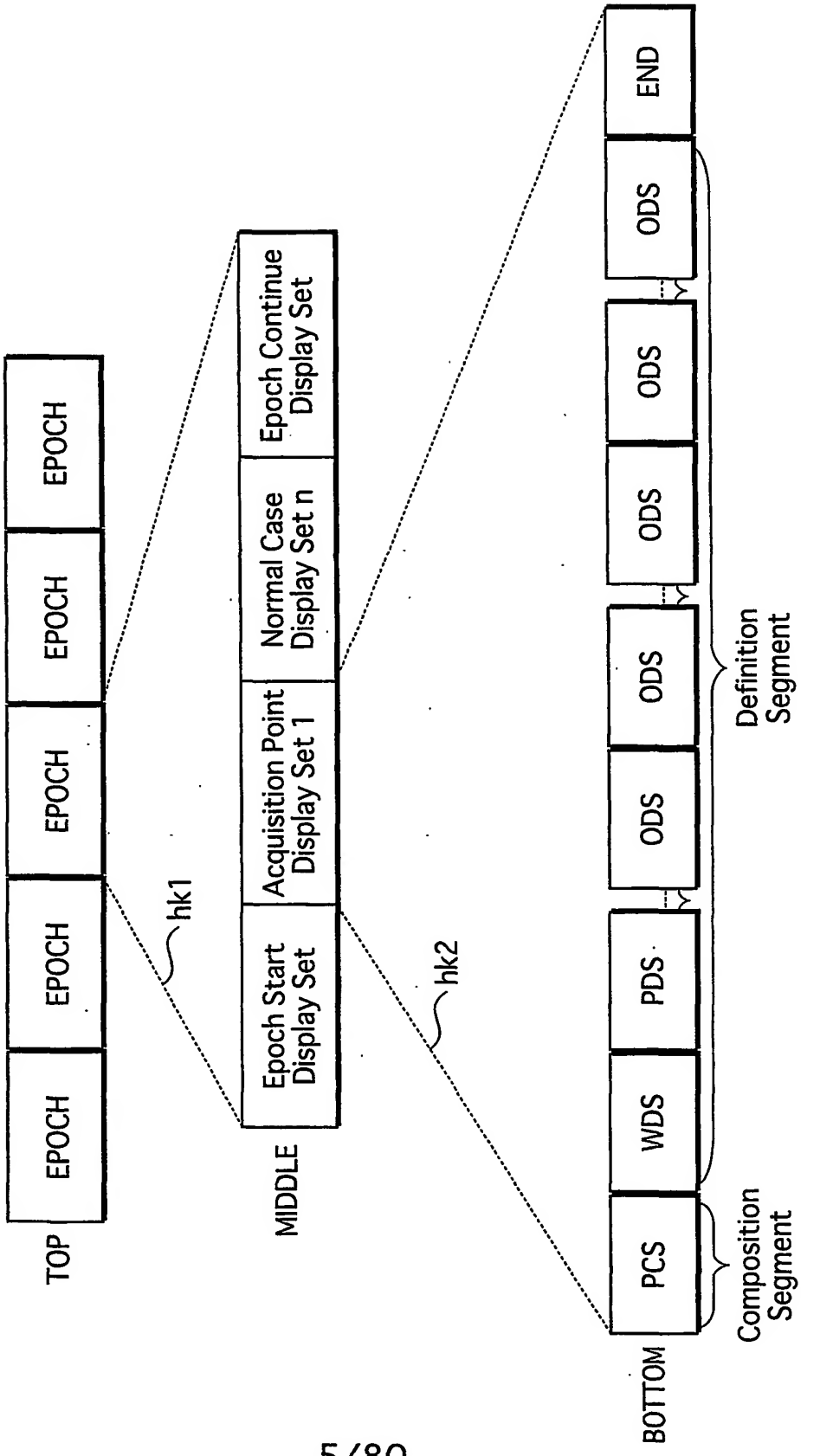

PES Packet
PTS
DTS
PCS
WDS, PDS, ODS
PACKET HEADER
PAYLOAD

FIG. 4B

FIG. 5


TOP
EPOCH
EPOCH
EPOCH
EPOCH
EPOCH
hk1
MIDDLE
Epoch Start Display Set
Acquisition Point Display Set 1
Normal Case Display Set n
Epoch Continue Display Set
hk2
BOTTOM
PCS
WDS
PDS
ODS
ODS
ODS
ODS
ODS
END
Composition Segment
Definition Segment

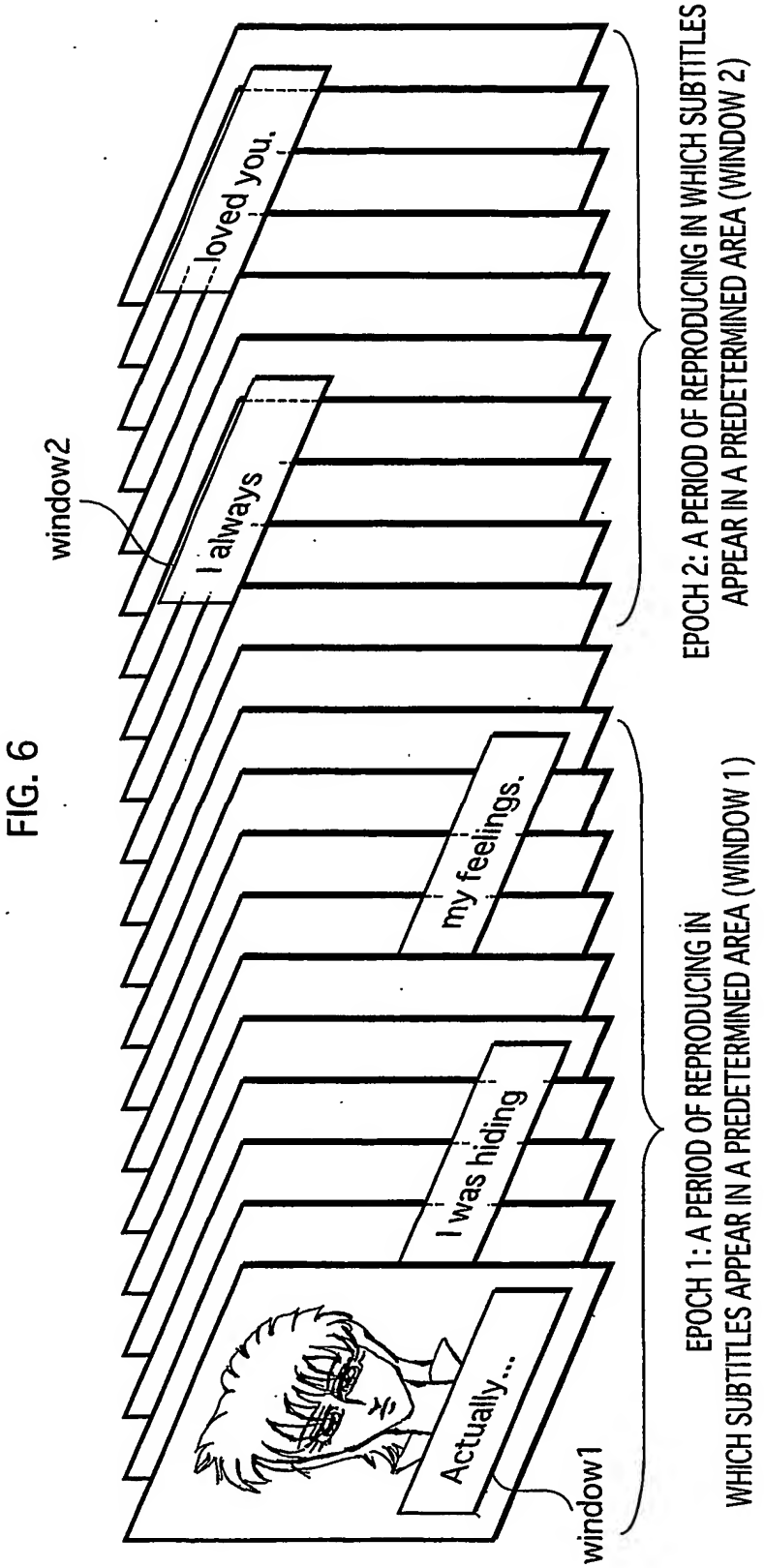

loved you.
I always
my feelings.
I was hiding
Actually...
window2
window1
EPOCH 2: A PERIOD OF REPRODUCING IN WHICH SUBTITLES APPEAR IN A PREDETERMINED AREA (WINDOW 2)
EPOCH 1: A PERIOD OF REPRODUCING IN WHICH SUBTITLES APPEAR IN A PREDETERMINED AREA (WINDOW 1)

FIG. 6

FIG. 7A

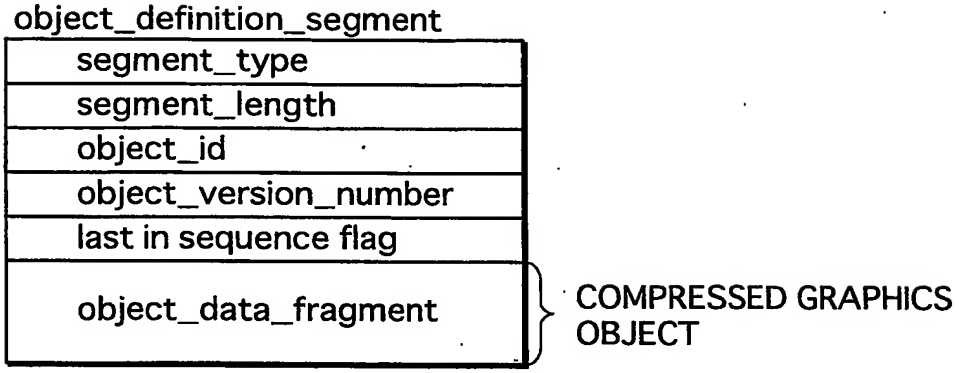

object_definition_segment
segment_type
segment_length
object_id
object_version_number
last in sequence flag
object_data_fragment
COMPRESSED GRAPHICS OBJECT

FIG. 7B

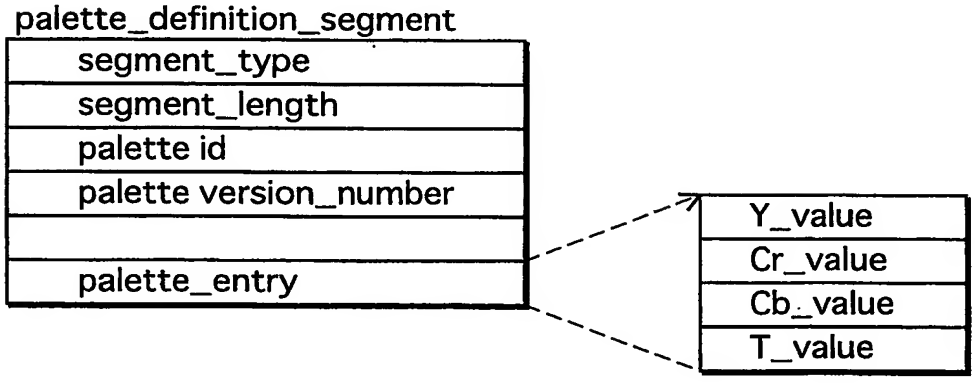

palette_definition_segment
segment_type
segment_length
palette id
palette version_number
palette_entry
Y_value
Cr_value
Cb_value
T_value

FIG. 8A

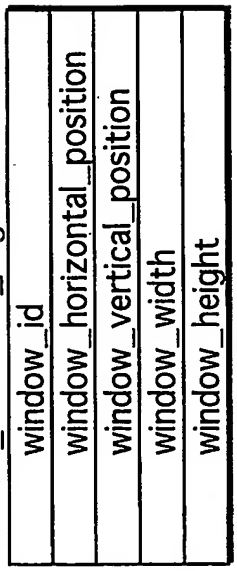

window_definition_segment
window_id
window_horizontal_position
window_vertical_position
window_width
window_height

FIG. 8B

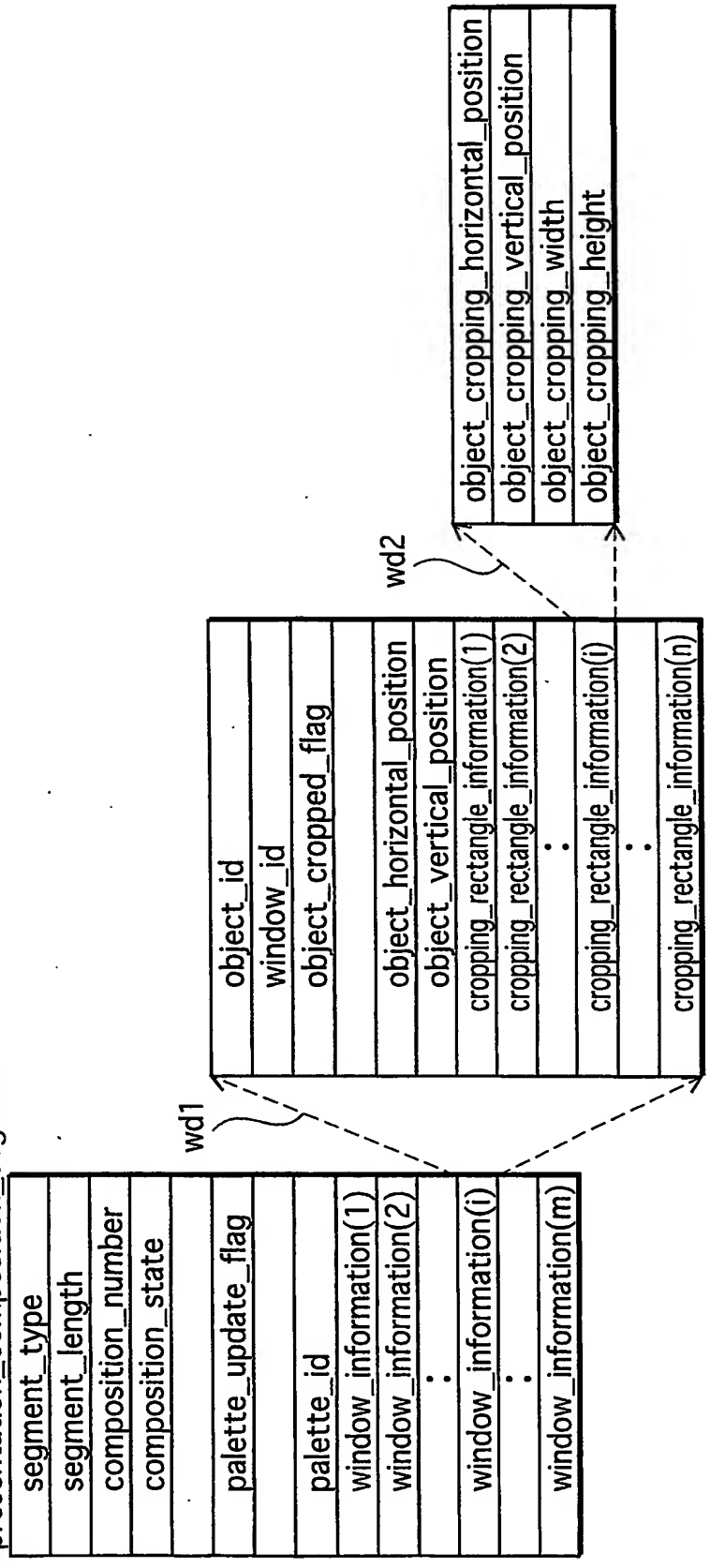

presentation_composition_segment
segment_type
segment_length
composition_number
composition_state
palette_update_flag
palette_id
window_information(1)
window_information(2)
window_information(i)
window_information(m)
wd1
object_id
window_id
object_cropped_flag
object_horizontal_position
object_vertical_position
cropping_rectangle_information(1)
cropping_rectangle_information(2)
cropping_rectangle_information(i)
cropping_rectangle_information(n)
wd2
object_cropping_horizontal_position
object_cropping_vertical_position
object_cropping_width
object_cropping_height

FIG. 9

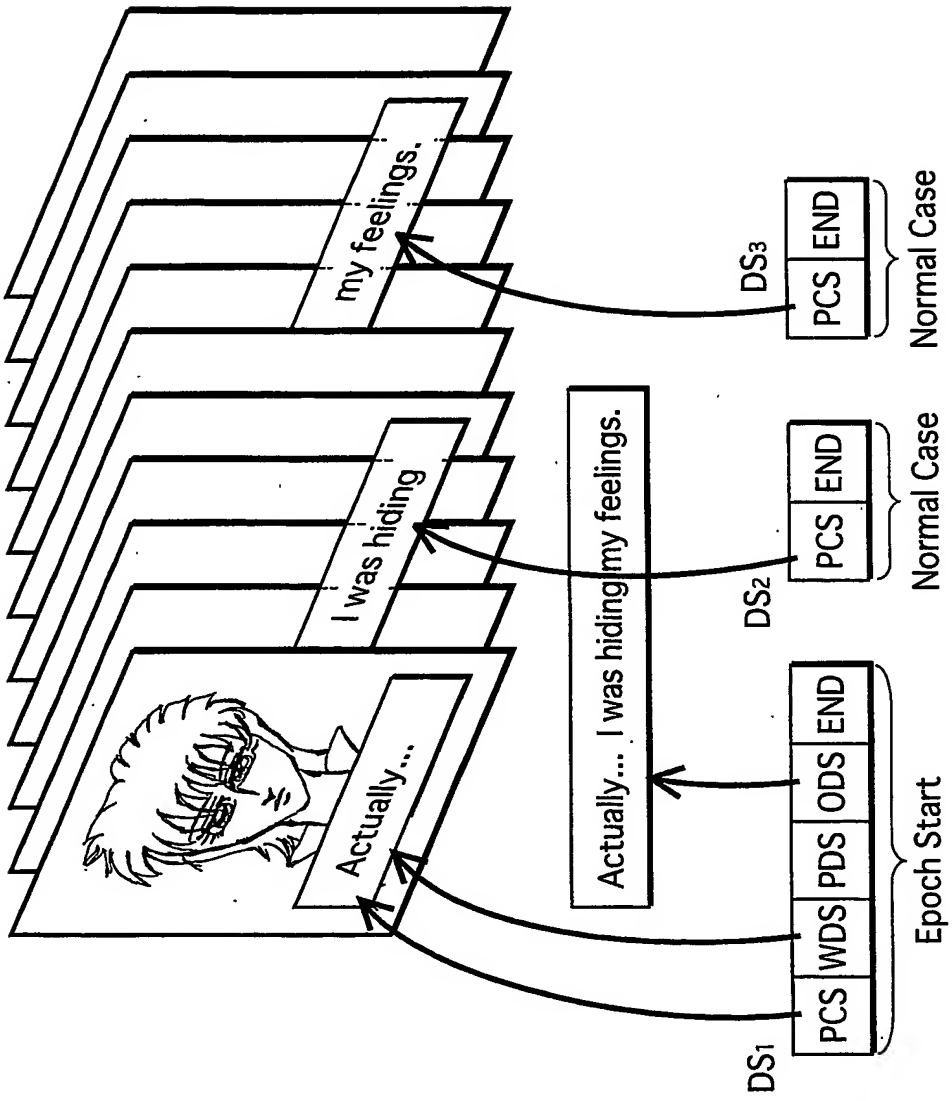

Actually...
I was hiding
my feelings.
Actually... I was hiding my feelings.
DS1
PCS
WDS
PDS
ODS
END
Epoch Start
DS2
PCS
END
Normal Case
DS3
PCS
END
Normal Case

FIG. 10

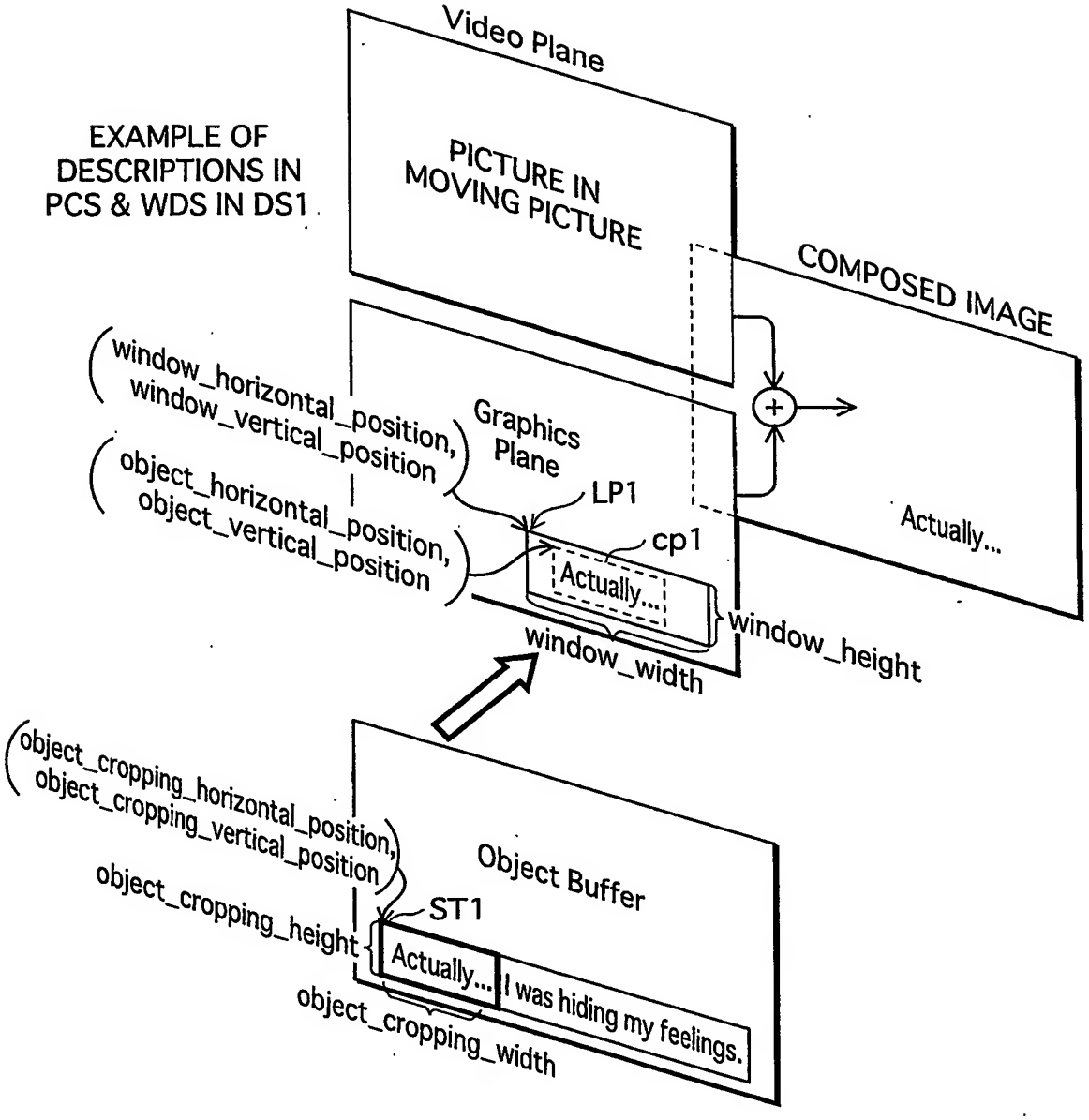

Video Plane
EXAMPLE OF DESCRIPTIONS IN PCS & WDS IN DS1
PICTURE IN MOVING PICTURE
COMPOSED IMAGE
window_horizontal_position,
window_vertical_position
Graphics Plane
object_horizontal_position,
object_vertical_position
LP1
cp1
Actually...
Actually...
window_height
window_width
object_cropping_horizontal_position,
object_cropping_vertical_position
Object Buffer
ST1
object_cropping_height
Actually...
I was hiding my feelings.
object_cropping_width

FIG. 11

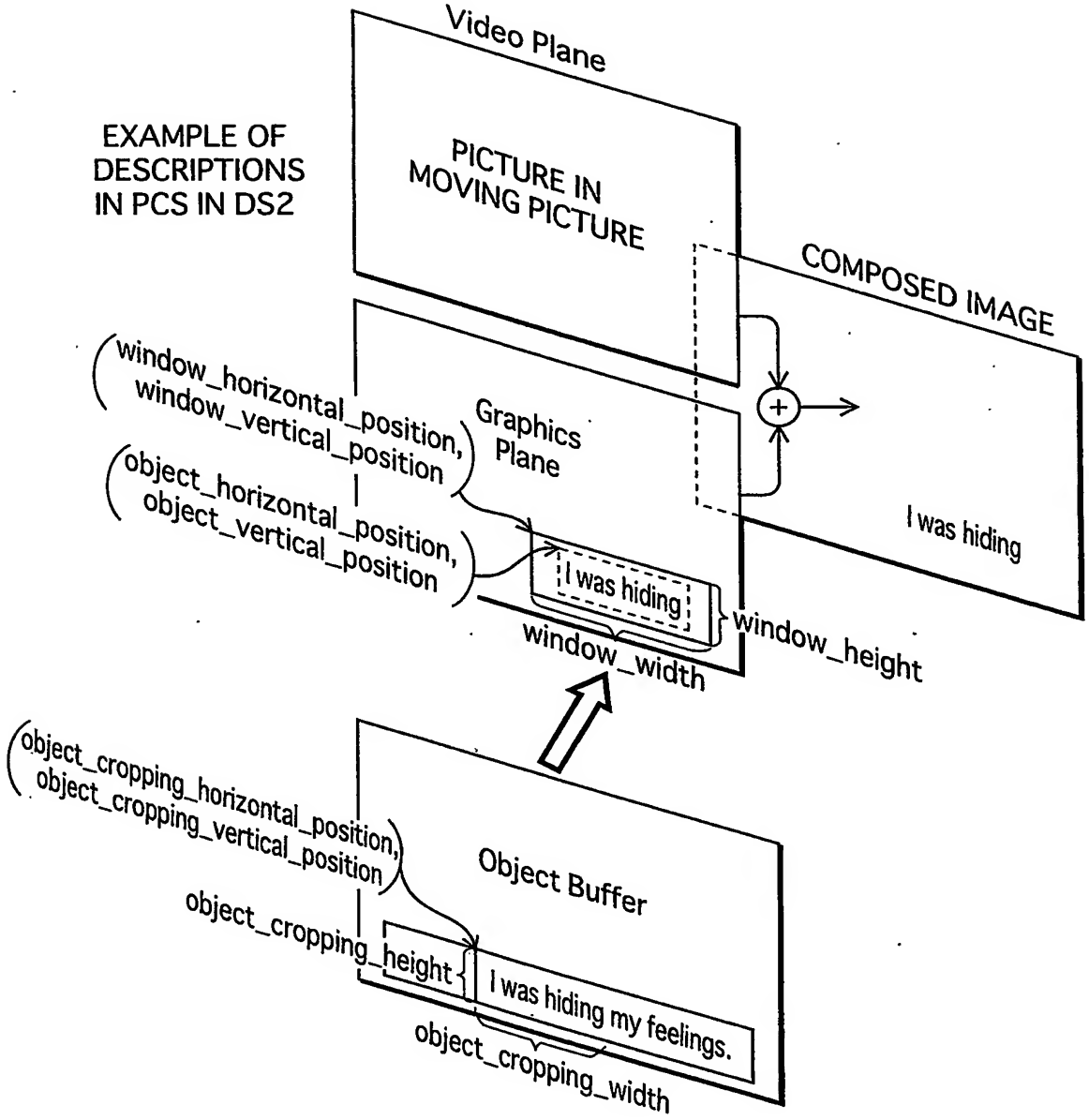

Video Plane
EXAMPLE OF DESCRIPTIONS IN PCS IN DS2
PICTURE IN MOVING PICTURE
COMPOSED IMAGE
window_horizontal_position,
window_vertical_position
Graphics Plane
object_horizontal_position,
object_vertical_position
I was hiding
I was hiding
window_height
window_width
object_cropping_horizontal_position,
object_cropping_vertical_position
Object Buffer
object_cropping_height
I was hiding my feelings.
object_cropping_width

FIG. 12

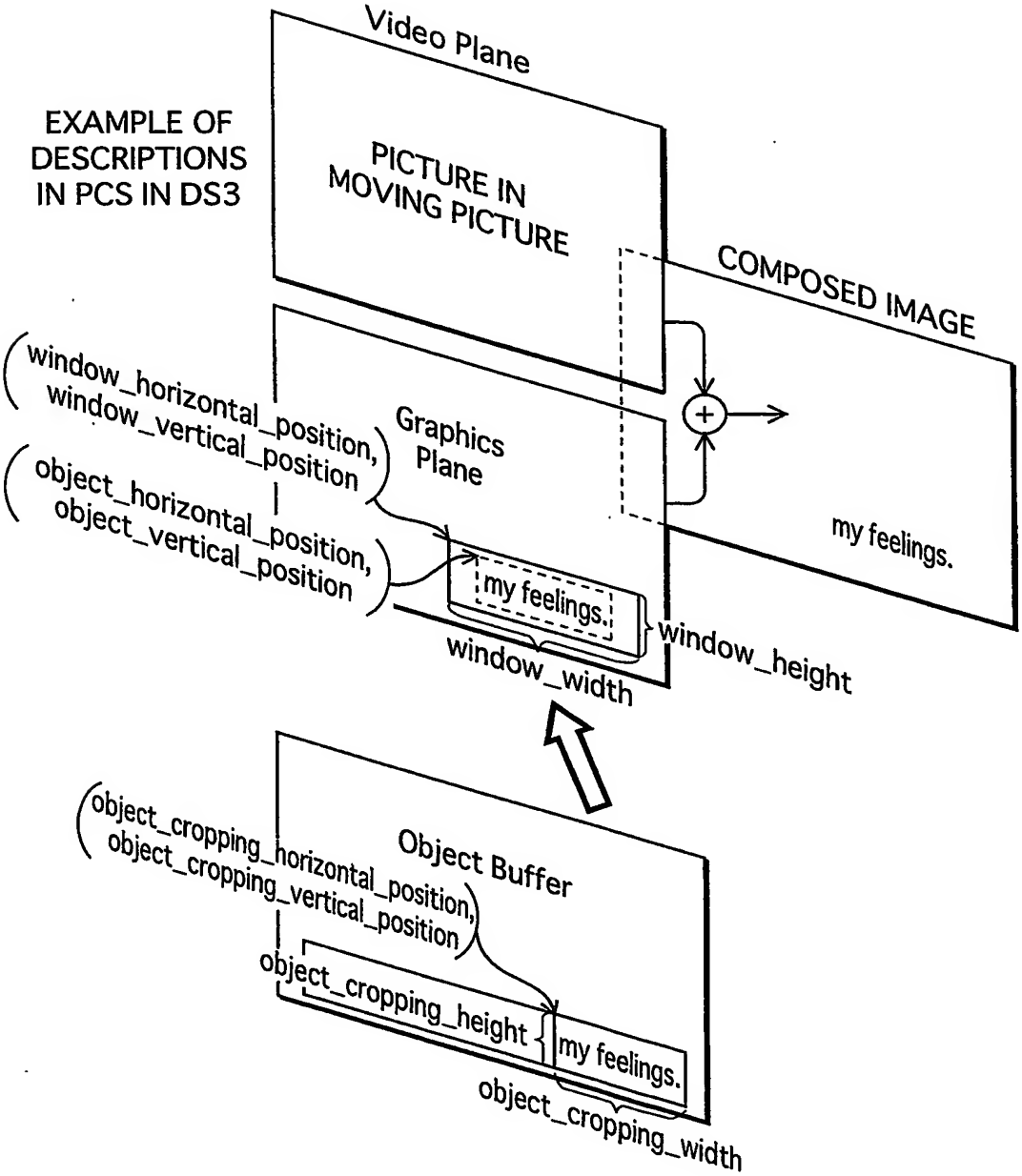

Video Plane
EXAMPLE OF DESCRIPTIONS IN PCS IN DS3
PICTURE IN MOVING PICTURE
COMPOSED IMAGE
window_horizontal_position,
window_vertical_position
Graphics Plane
object_horizontal_position,
object_vertical_position
my feelings.
my feelings.
window_height
window_width
Object Buffer
object_cropping_horizontal_position,
object_cropping_vertical_position
object_cropping_height
my feelings.
object_cropping_width

FIG. 13

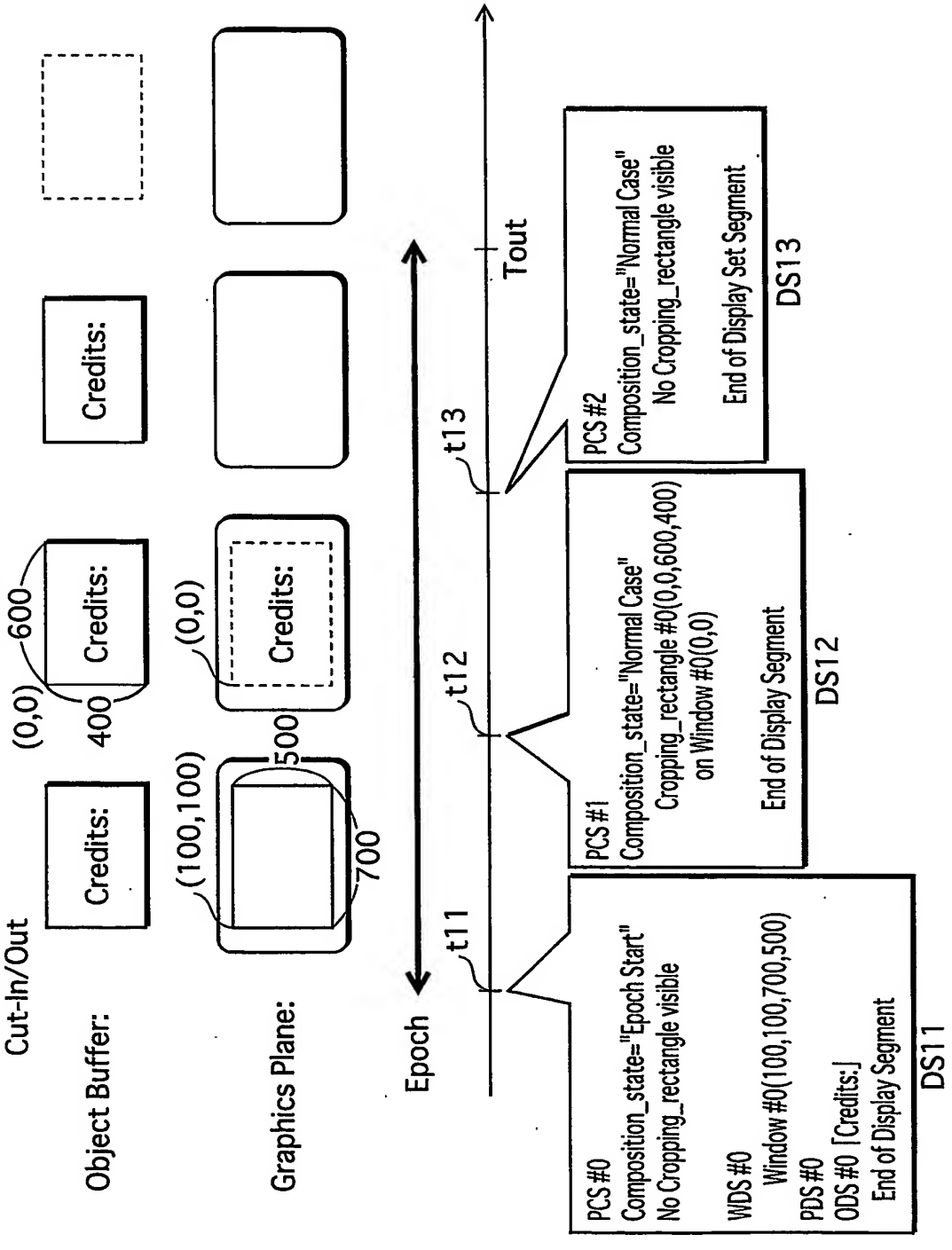

Cut-In/Out
Object Buffer:
Credits:
(0,0)
600
400
Credits:
Credits:
Graphics Plane:
(100,100)
500
700
(0,0)
Credits:
Epoch
t11
t12
t13
Tout
PCS #0
Composition_state="Epoch Start"
No Cropping_rectangle visible
WDS #0
Window #0(100,100,700,500)
PDS #0
ODS #0 [Credits:]
End of Display Segment
DS11
PCS #1
Composition_state="Normal Case"
Cropping_rectangle #0(0,0,600,400)
on Window #0(0,0)
End of Display Segment
DS12
PCS #2
Composition_state="Normal Case"
No Cropping_rectangle visible
End of Display Set Segment
DS13

FIG. 14

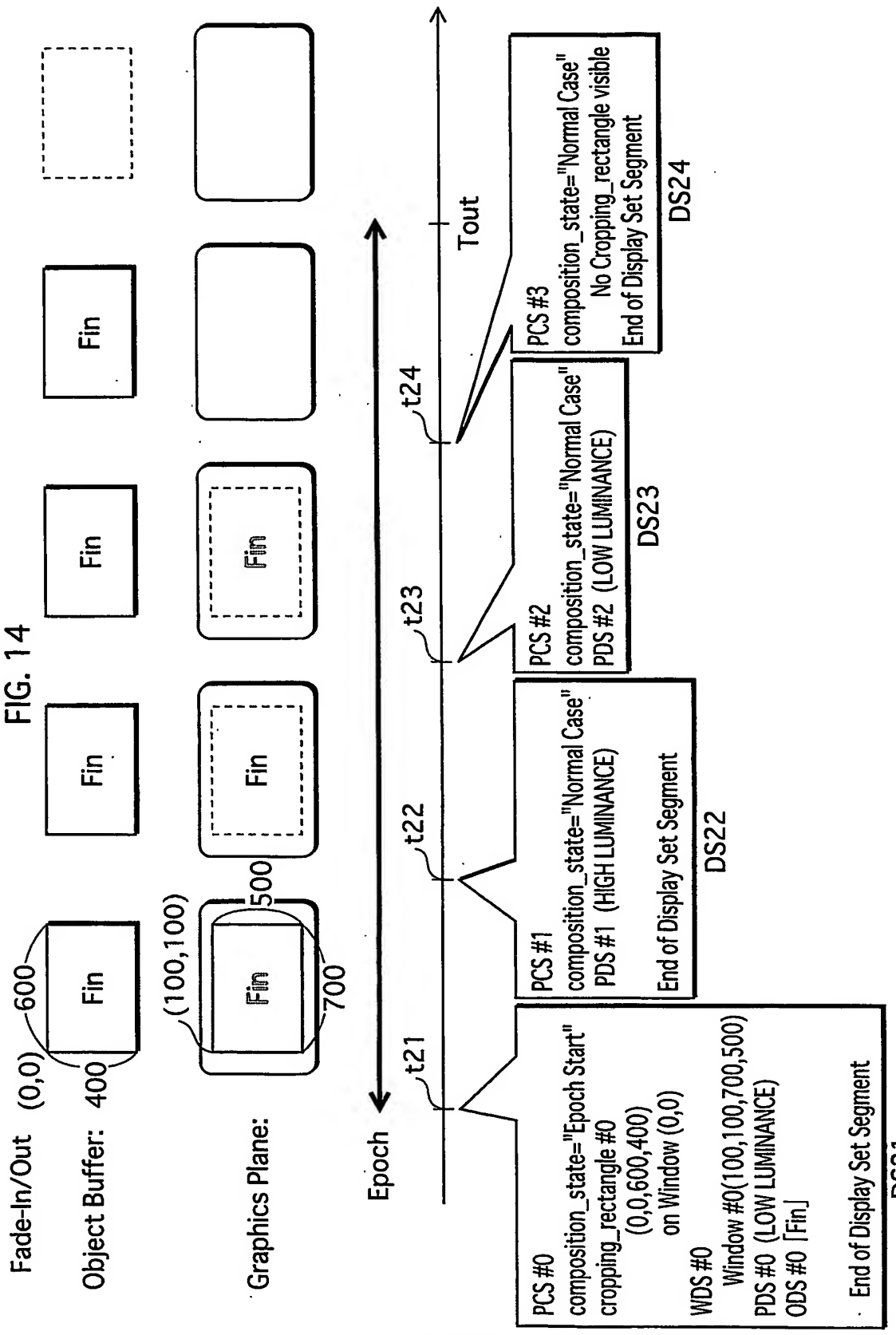

Fade-In/Out
Object Buffer:
Graphics Plane:
(0,0)
600
400
(100,100)
500
700
Fin
Epoch
t21
t22
t23
t24
Tout
PCS #0
composition_state="Epoch Start"
cropping_rectangle #0
(0,0,600,400)
on Window (0,0)
WDS #0
Window #0(100,100,700,500)
PDS #0 (LOW LUMINANCE)
ODS #0 [Fin]
End of Display Set Segment
DS21
PCS #1
composition_state="Normal Case"
PDS #1 (HIGH LUMINANCE)
End of Display Set Segment
DS22
PCS #2
composition_state="Normal Case"
PDS #2 (LOW LUMINANCE)
DS23
PCS #3
composition_state="Normal Case"
No Cropping_rectangle visible
End of Display Set Segment
DS24

FIG. 15

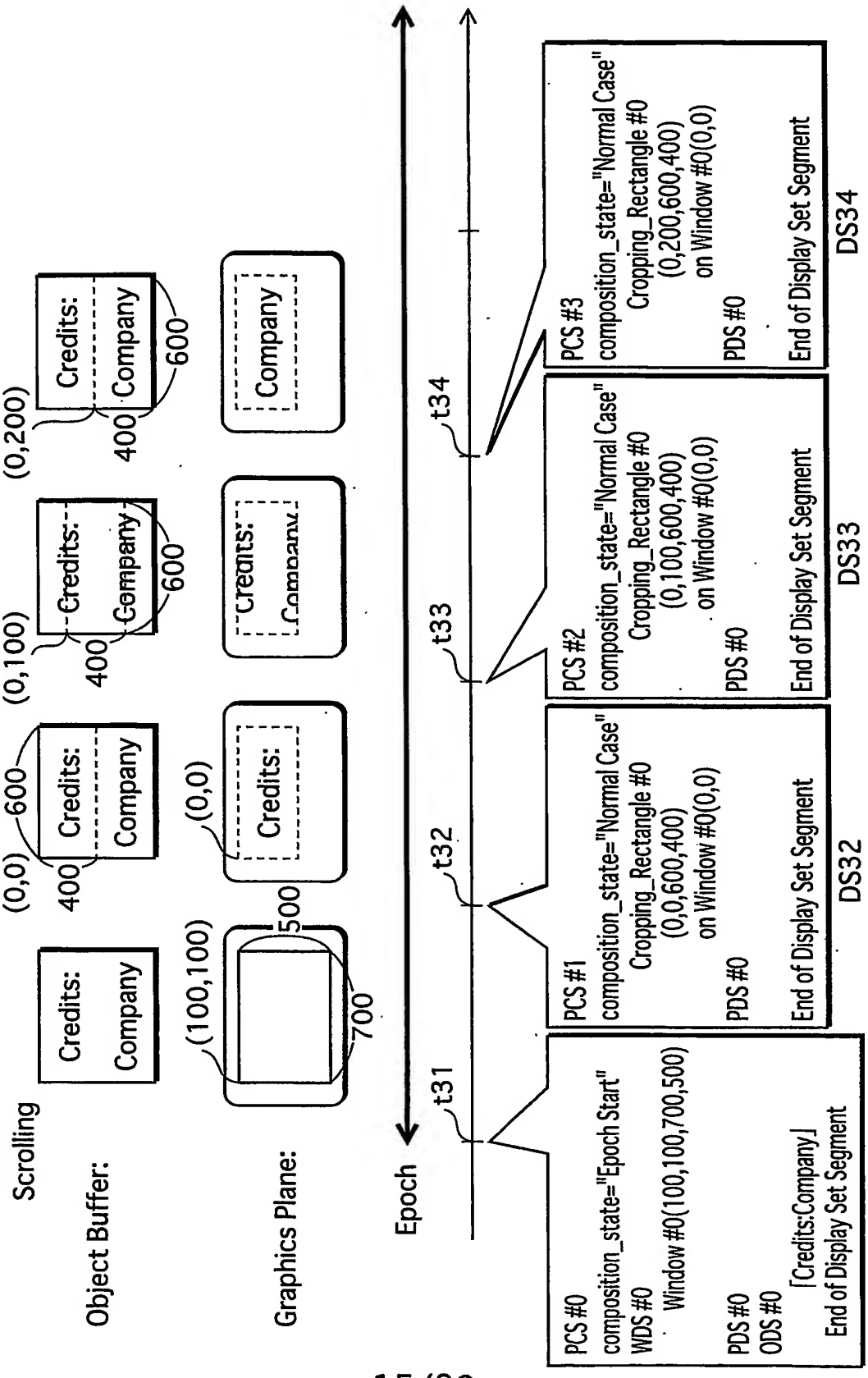

Scrolling
Object Buffer:
Graphics Plane:
Credits:
Company
(0,0)
400
600
(100,100)
500
700
(0,100)
(0,200)
Epoch
t31
t32
t33
t34
PCS #0
composition_state="Epoch Start"
WDS #0
Window #0(100,100,700,500)
PDS #0
ODS #0
「Credits:Company」
End of Display Set Segment
DS31
PCS #1
composition_state="Normal Case"
Cropping_Rectangle #0
(0,0,600,400)
on Window #0(0,0)
PDS #0
End of Display Set Segment
DS32
PCS #2
composition_state="Normal Case"
Cropping_Rectangle #0
(0,100,600,400)
on Window #0(0,0)
PDS #0
End of Display Set Segment
DS33
PCS #3
composition_state="Normal Case"
Cropping_Rectangle #0
(0,200,600,400)
on Window #0(0,0)
PDS #0
End of Display Set Segment
DS34

FIG. 16

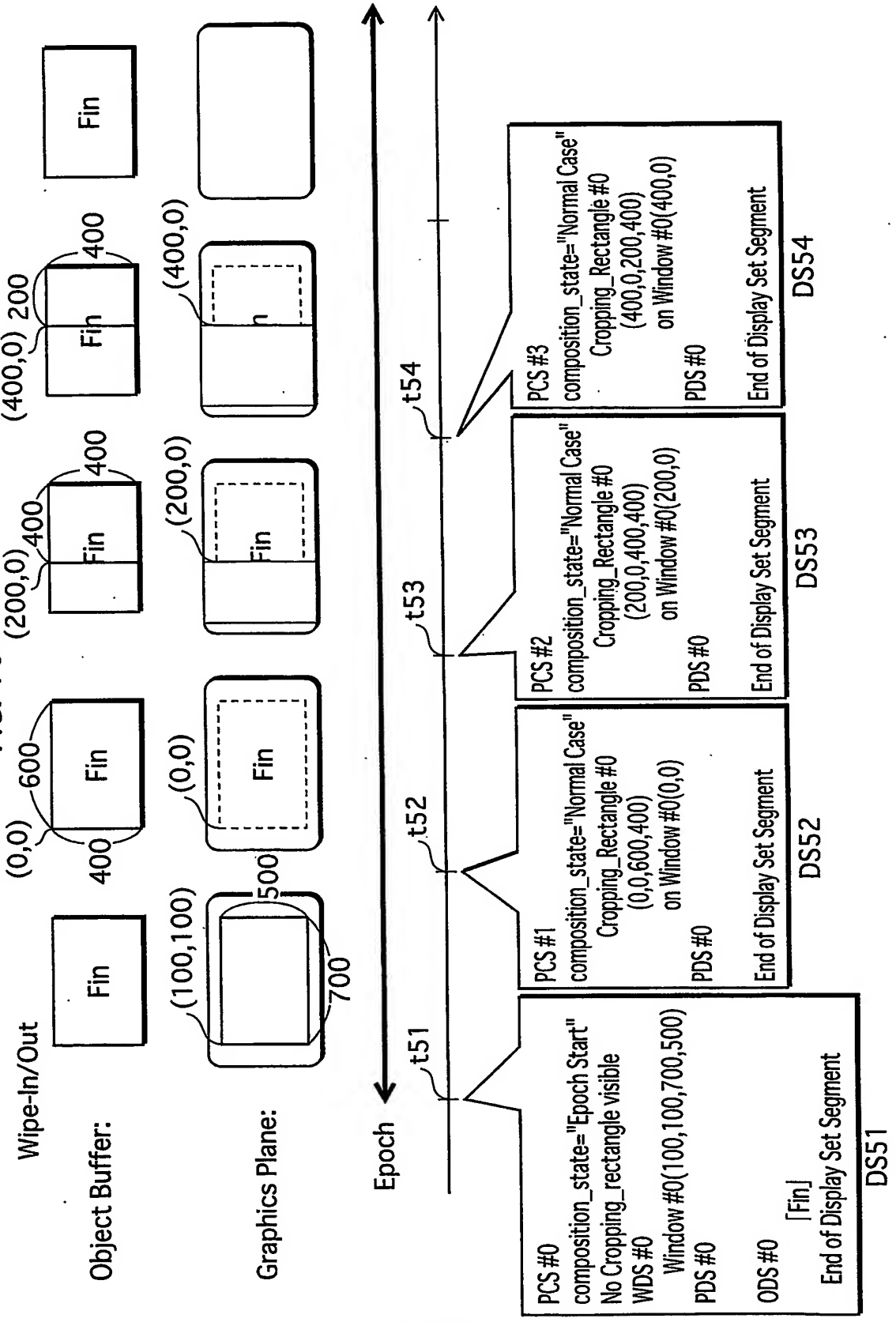

Wipe-In/Out
Object Buffer:
Graphics Plane:
Epoch
Fin
(0,0)
600
400
(100,100)
500
700
(200,0)
400
(400,0)
200
(0,0)
(200,0)
(400,0)
t51
t52
t53
t54
PCS #0
composition_state="Epoch Start"
No Cropping_rectangle visible
WDS #0
Window #0(100,100,700,500)
PDS #0
ODS #0
[Fin]
End of Display Set Segment
DS51
PCS #1
composition_state="Normal Case"
Cropping_Rectangle #0
(0,0,600,400)
on Window #0(0,0)
PDS #0
End of Display Set Segment
DS52
PCS #2
composition_state="Normal Case"
Cropping_Rectangle #0
(200,0,400,400)
on Window #0(200,0)
PDS #0
End of Display Set Segment
DS53
PCS #3
composition_state="Normal Case"
Cropping_Rectangle #0
(400,0,200,400)
on Window #0(400,0)
PDS #0
End of Display Set Segment
DS54

FIG. 17

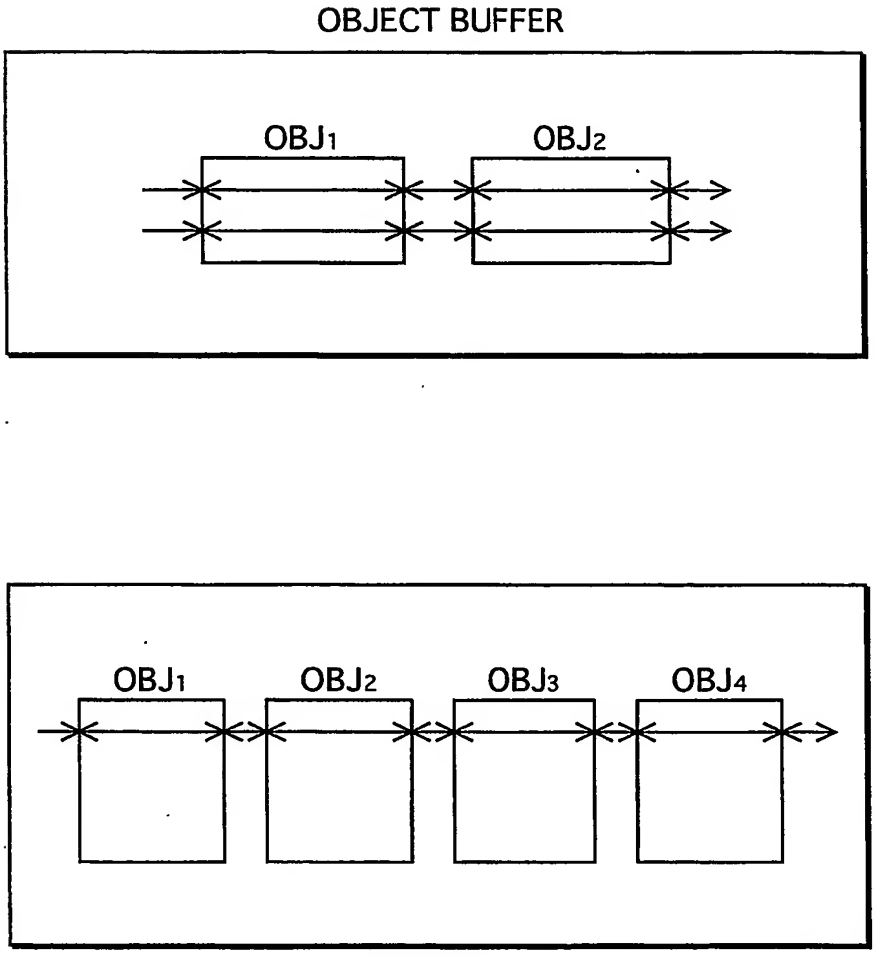


FIG. 18

```
PTS( DSn[PCS)] )>=DTS( DSn[PCS] )+DECODEDURATION( DSn )
Where:
  • DECODEDURATION( DSn ) is calculated as follows:
decode_duration = 0 ;
decode_duration += PLANEINITIALIZATIONTIME( DSn ) ;
if( DSn. PCS. num_of_objects == 2 )
{
    decode_duration += WAIT( DSn, DSn. PCS. OBJ[0], decode_duration ) ;
    if( DSn. PCS. OBJ[0]. window_id == DSn. PCS. OBJ[1]. window_id )
    {
            decode_duration += WAIT( DSn, DSn. PCS. OBJ[1], decode_duration ) ;
            decode_duration += 90000*( SIZE( DSn. PCS. OBJ[0]. window_id )//256*10^6 ) ;
    }
    else
    {
            decode_duration += 90000*( SIZE( DSn. PCS. OBJ[0]. window_id )//256*10^6 ) ;
            decode_duration += WAIT( DSn, DSn. PCS. OBJ[1], decode_duration );
            decode_duration += 90000*( SIZE( DSn. PCS. OBJ[1]. window_id )//256*10^6 ) ;
    }
}
else if( DSn. PCS. num_of_objects ==1 )
{
    decode_duration += WAIT( DSn, DSn. PCS. OBJ[0], decode_duration ) ;
    decode_duration += 90000*( SIZE( DSn. PCS. OBJ[0]. window_id )//256*10^6 ) ;
}
return decode_duration ;

  • PLANEINITIALIZATIONTIME( DSn ) is calculated as follows:
initialize_duration=0 ;
if( DSn. PCS. composition_state= = EPOCH_START )
{
  initialize_duration = 90000*( 8*video_width*video_height//256*10^6) ;
}

else
 {
    for( i=0 ; i < WDS. num_windows ; i++ )
    {
            if( EMPTY(DSn.WDS.WIN[i],DSn ) )
                    initialize_duration += 90000*( SIZE( DSn. WDS. WIN[i] )//256*10^6) ;
    }
}
return initialize_duration ;

  • WAIT( DSn, OBJ, current_duration ) is calculated as follows:
wait_duration = 0 ;
if( EXISTS( OBJ. object_id, DSn ) )
{
    object_definition_ready_time = PTS( GET( OBJ. object_id. DSn ) ) ;
    current_time = DTS( DSn. PCS )+current_duration ;
    if( current_time < object_definition_ready_time )
            wait_duration += object_definition_ready_time - current_time ) ;
}
return wait_duration ;
```

FIG. 19

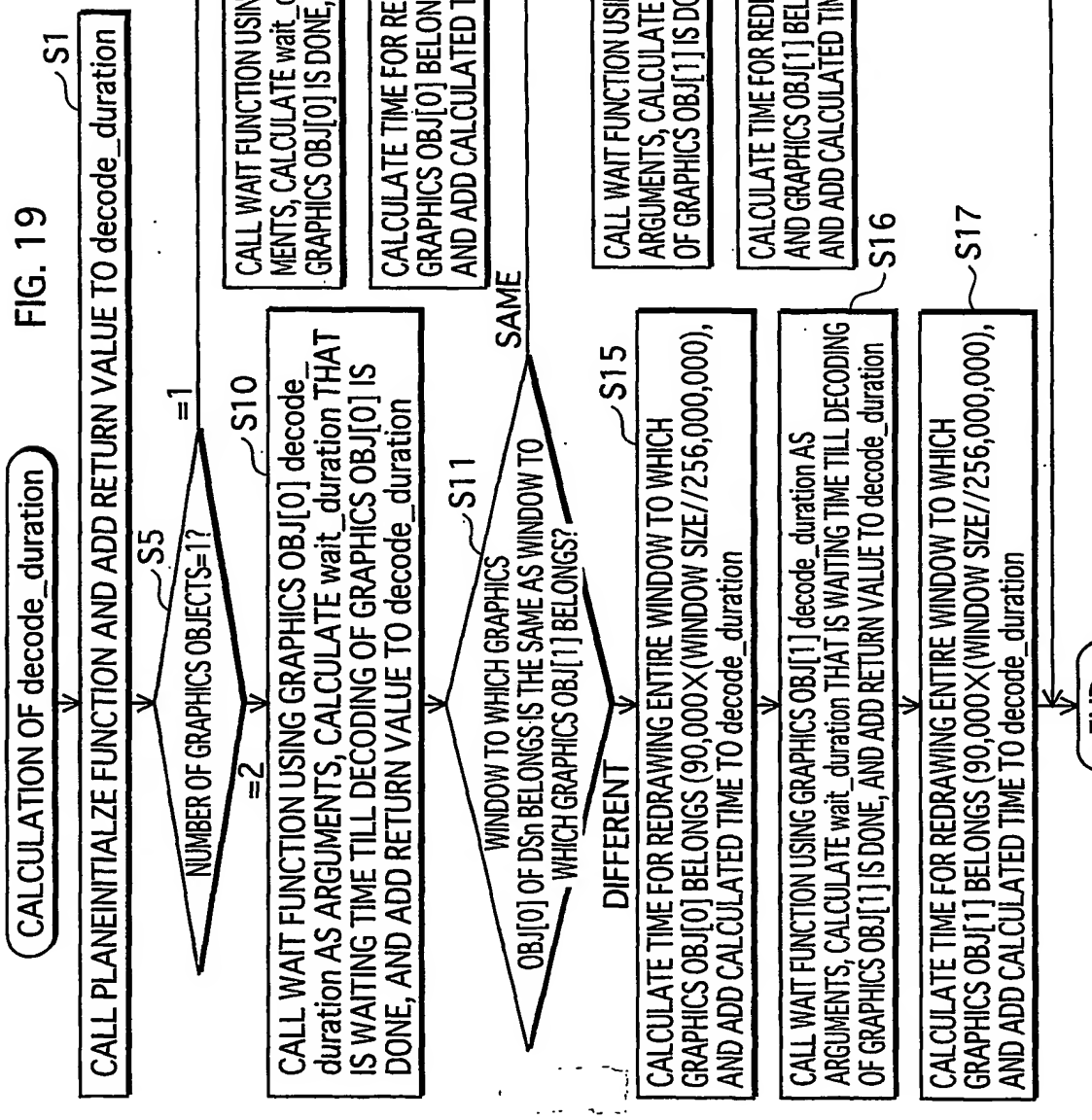

CALCULATION OF decode_duration
CALL PLANEINITIALZE FUNCTION AND ADD RETURN VALUE TO decode_duration
S1
S5
NUMBER OF GRAPHICS OBJECTS=1?
=1
=2
S6
CALL WAIT FUNCTION USING GRAPHICS OBJ[0] decode_duration AS ARGUMENTS, CALCULATE wait_duration THAT IS WAITING TIME TILL DECODING OF GRAPHICS OBJ[0] IS DONE, AND ADD RETURN VALUE TO decode_duration
S9
CALCULATE TIME FOR REDRAWING ENTIRE WINDOW TO WHICH GRAPHICS OBJ[0] BELONGS (90,000×(WINDOW SIZE//256,000,000), AND ADD CALCULATED TIME TO decode_duration
S10
CALL WAIT FUNCTION USING GRAPHICS OBJ[0] decode_duration AS ARGUMENTS, CALCULATE wait_duration THAT IS WAITING TIME TILL DECODING OF GRAPHICS OBJ[0] IS DONE, AND ADD RETURN VALUE TO decode_duration
S11
WINDOW TO WHICH GRAPHICS OBJ[0] OF DSn BELONGS IS THE SAME AS WINDOW TO WHICH GRAPHICS OBJ[1] BELONGS?
SAME
DIFFERENT
A
S12
CALL WAIT FUNCTION USING GRAPHICS OBJ[1] decode_duration AS ARGUMENTS, CALCULATE wait_duration THAT IS WAITING TIME TILL DECODING OF GRAPHICS OBJ[1] IS DONE, AND ADD RETURN VALUE TO decode_duration
S13
CALCULATE TIME FOR REDRAWING ENTIRE WINDOWS TO WHICH GRAPHICS OBJ[0] AND GRAPHICS OBJ[1] BELONG (90,000×(WINDOW SIZE//256,000,000), AND ADD CALCULATED TIME TO decode_duration
S15
CALCULATE TIME FOR REDRAWING ENTIRE WINDOW TO WHICH GRAPHICS OBJ[0] BELONGS (90,000×(WINDOW SIZE//256,000,000), AND ADD CALCULATED TIME TO decode_duration
S16
CALL WAIT FUNCTION USING GRAPHICS OBJ[1] decode_duration AS ARGUMENTS, CALCULATE wait_duration THAT IS WAITING TIME TILL DECODING OF GRAPHICS OBJ[1] IS DONE, AND ADD RETURN VALUE TO decode_duration
S17
CALCULATE TIME FOR REDRAWING ENTIRE WINDOW TO WHICH GRAPHICS OBJ[1] BELONGS (90,000×(WINDOW SIZE//256,000,000), AND ADD CALCULATED TIME TO decode_duration
A
END

FIG. 20A

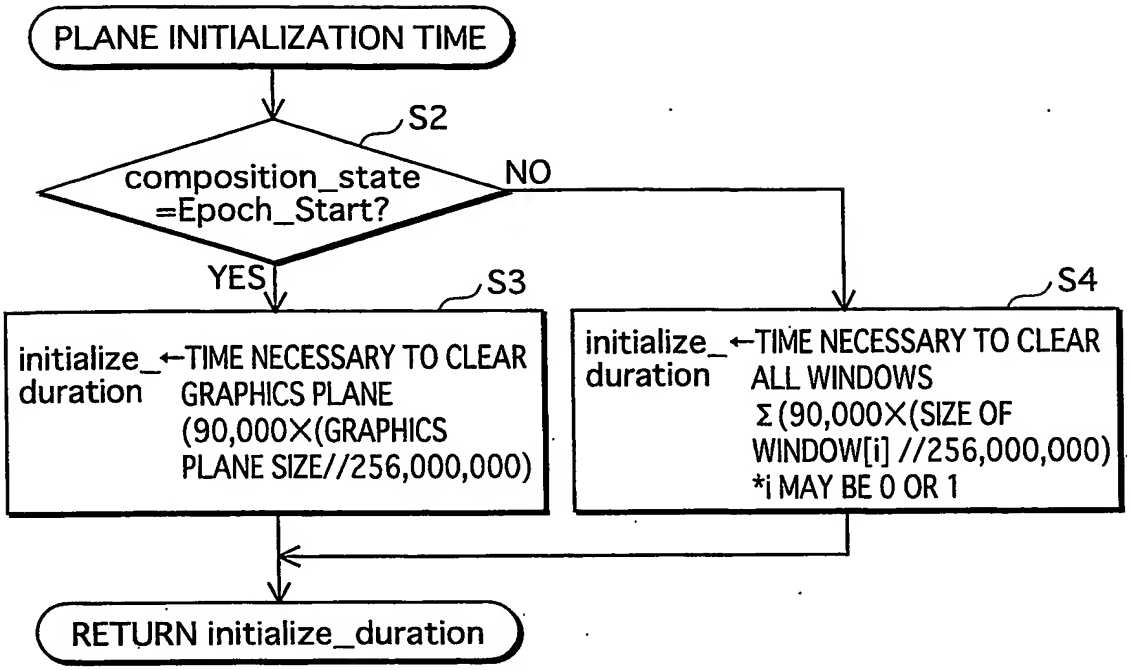


FIG. 20B

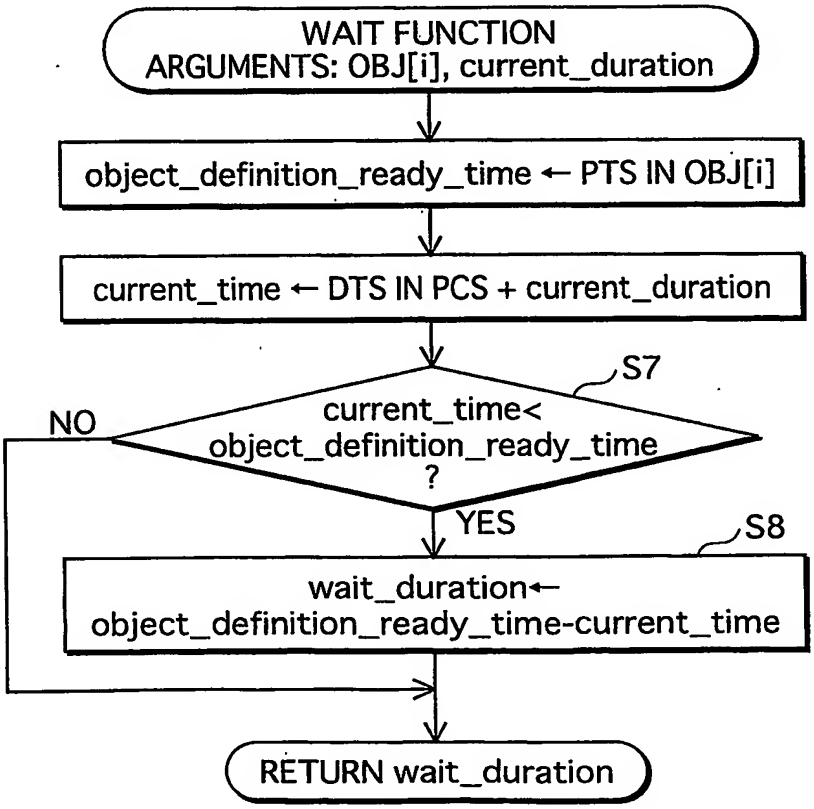


FIG. 21A

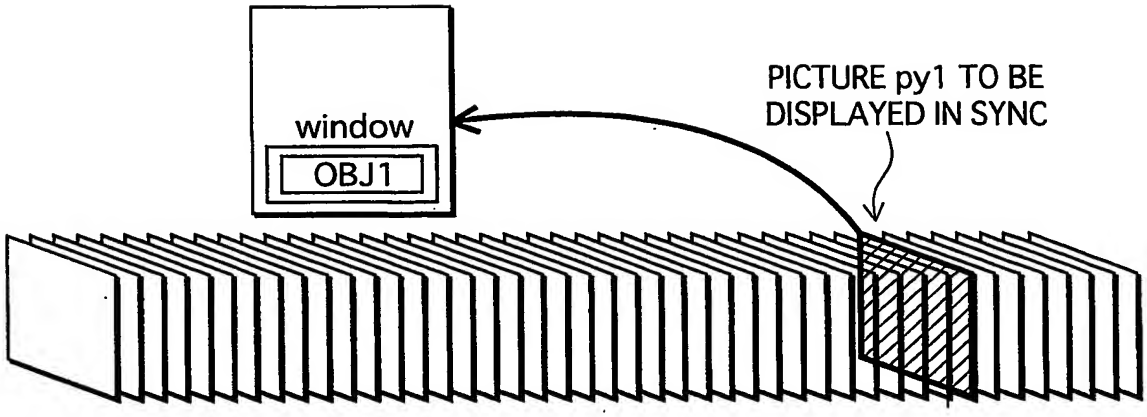


FIG. 21B

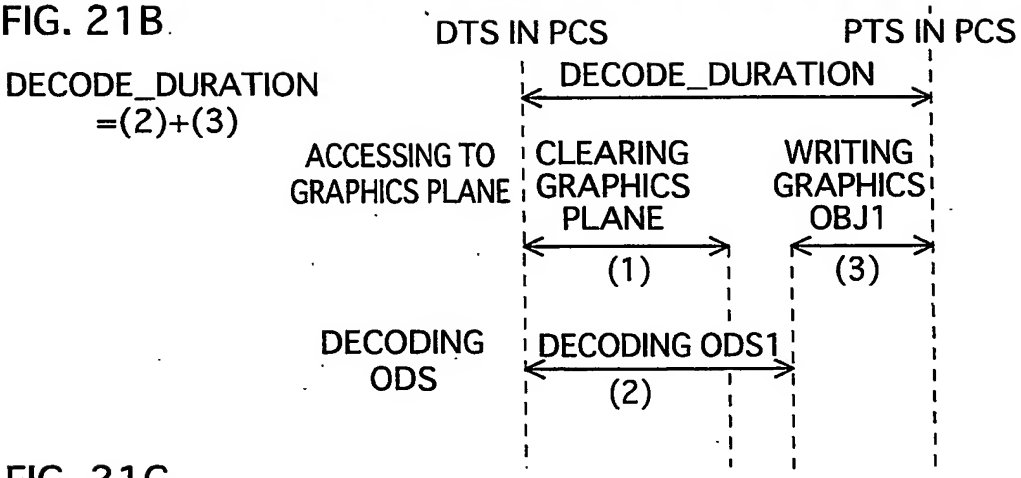


FIG. 21C

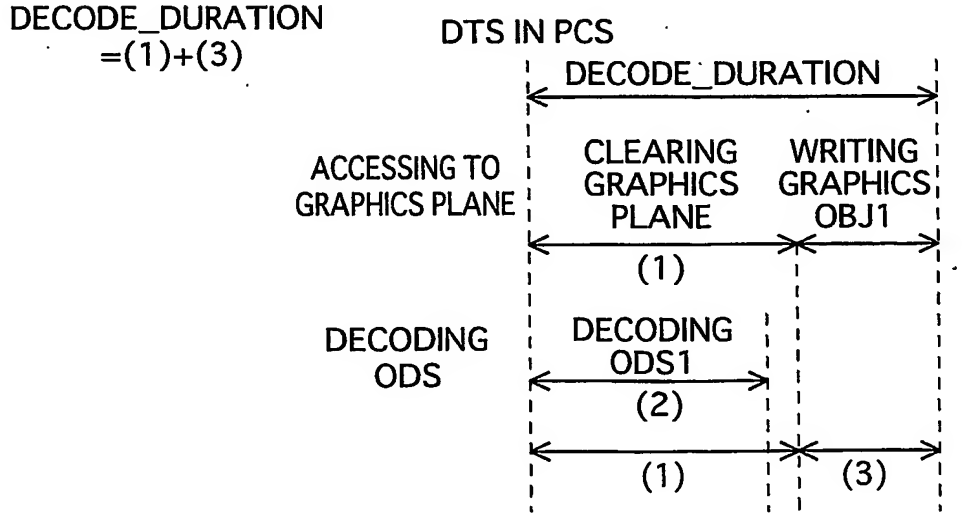


FIG. 22A

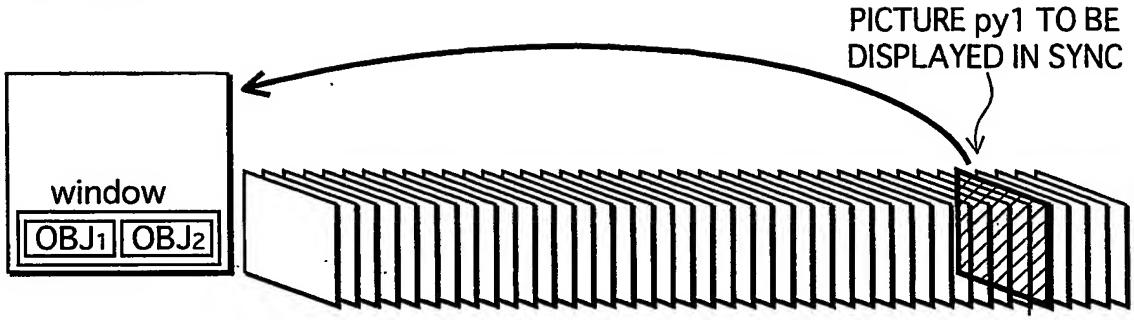


FIG. 22B

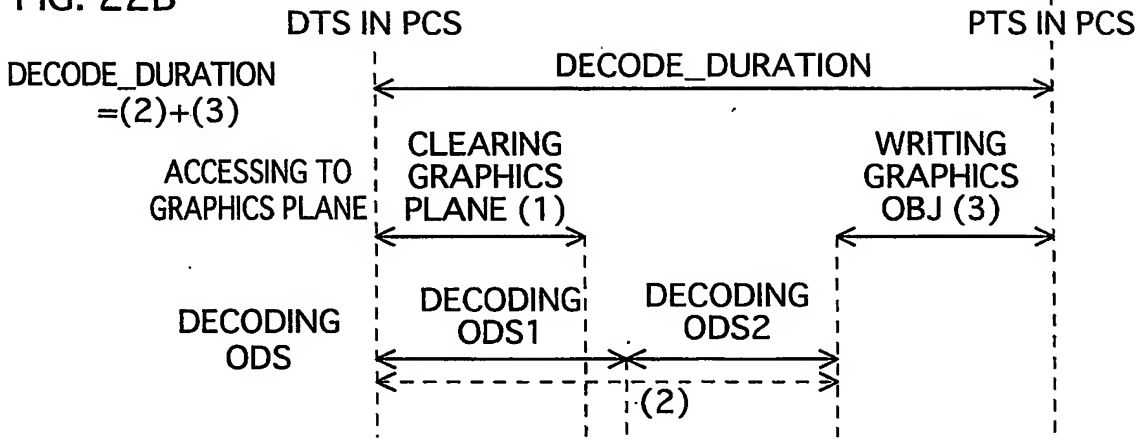
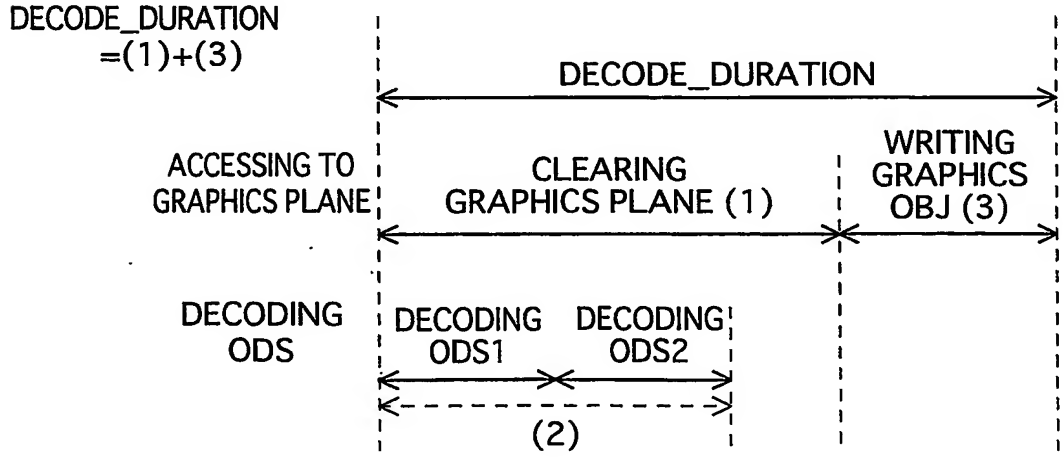


FIG. 22C



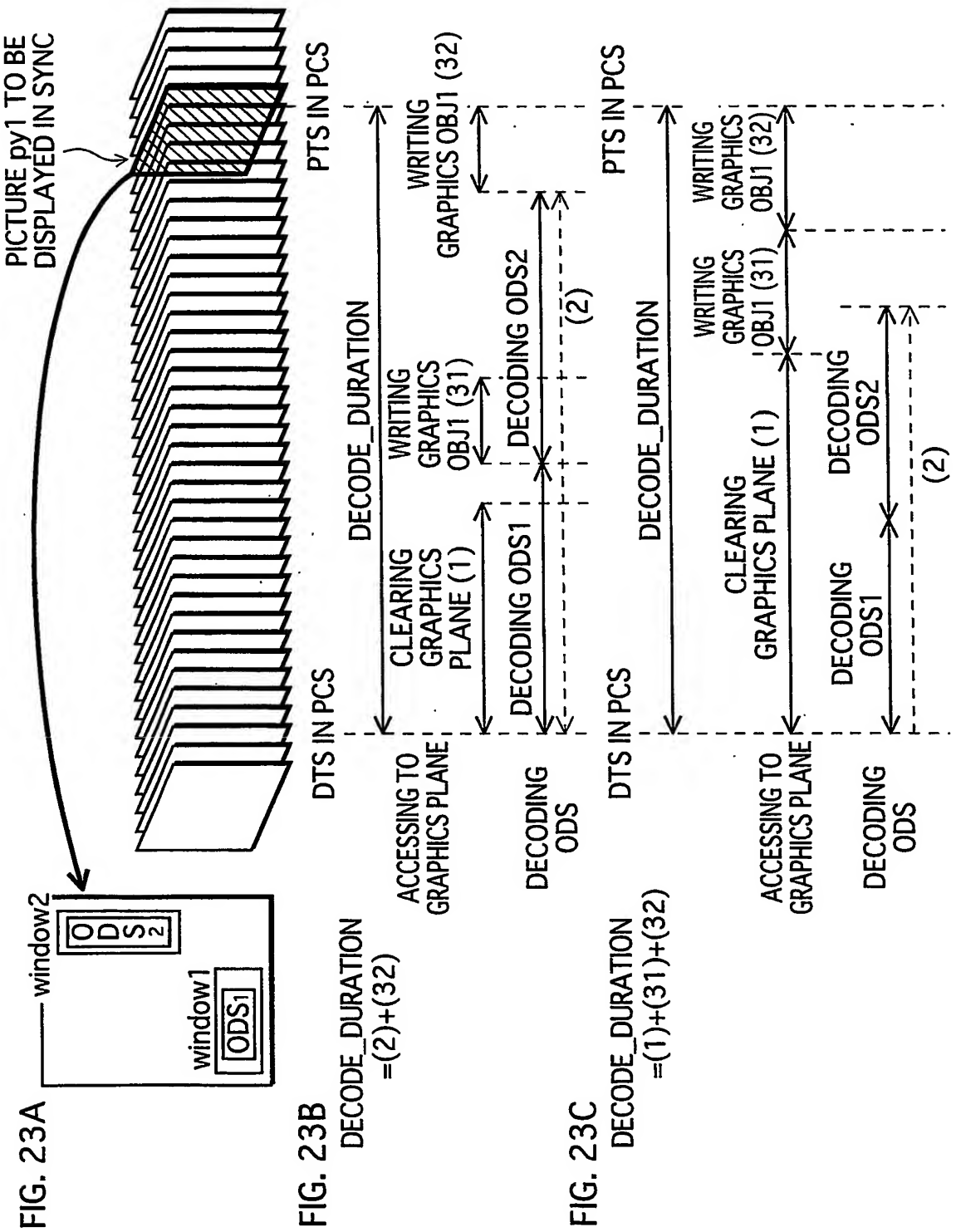

PICTURE py1 TO BE DISPLAYED IN SYNC
window2
ODS2
window1
ODS1
FIG. 23A
PTS IN PCS
DECODE_DURATION
WRITING GRAPHICS OBJ1 (32)
WRITING GRAPHICS OBJ1 (31)
DECODING ODS2
(2)
CLEARING GRAPHICS PLANE (1)
DECODING ODS1
DTS IN PCS
ACCESSING TO GRAPHICS PLANE
DECODING ODS
DECODE_DURATION =(2)+(32)
FIG. 23B
PTS IN PCS
DECODE_DURATION
WRITING GRAPHICS OBJ1 (32)
WRITING GRAPHICS OBJ1 (31)
CLEARING GRAPHICS PLANE (1)
DECODING ODS2
(2)
DECODING ODS1
DTS IN PCS
ACCESSING TO GRAPHICS PLANE
DECODING ODS
DECODE_DURATION =(1)+(31)+(32)
FIG. 23C

FIG. 24

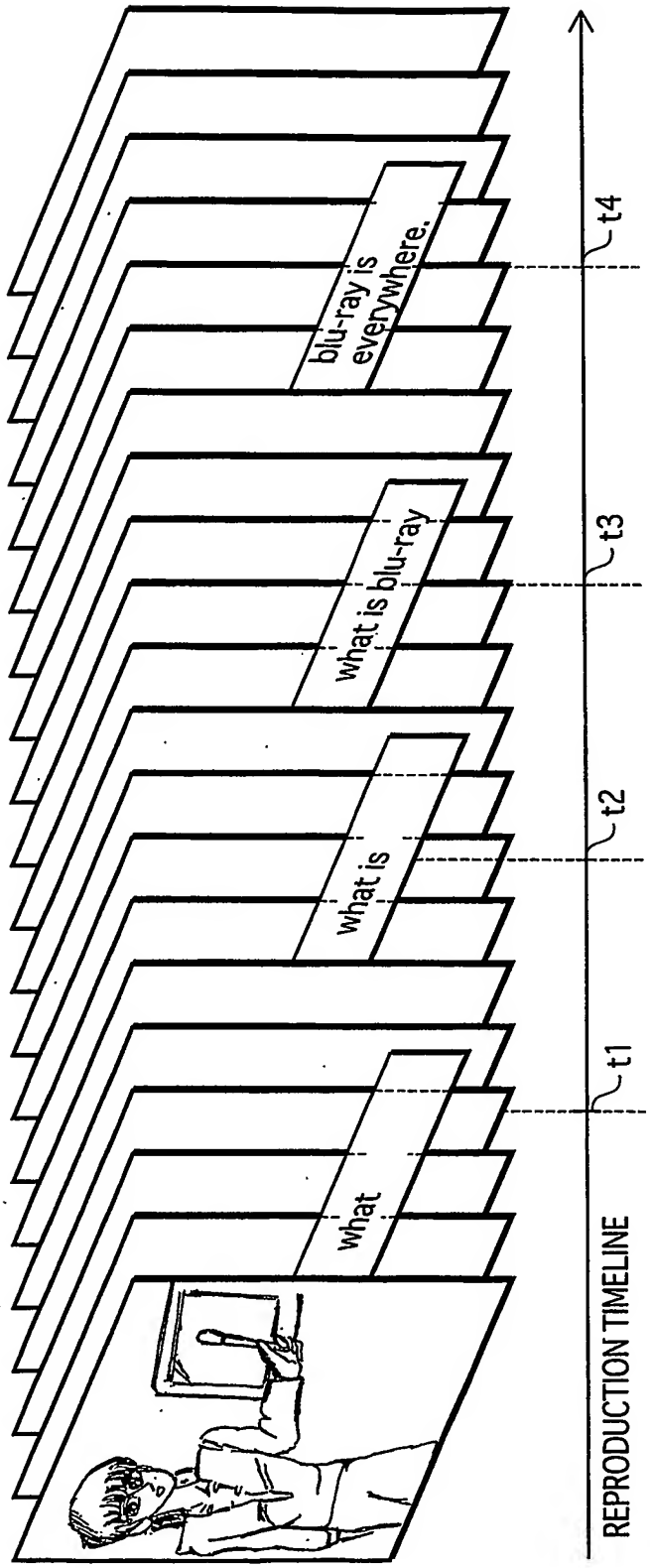

what
what is
what is blu-ray
blu-ray is everywhere.
t1
t2
t3
t4
REPRODUCTION TIMELINE

FIG. 25A

(DS1) PCS1.1-PDS1-ODS1-END
ODS1 "what is blu-ray"

(DS2) PCS1.2-END

(DS3) PCS1.3-END

(DS4) PCS2-ODS2-END
ODS2 " blu-ray is " everywhere

FIG. 25B

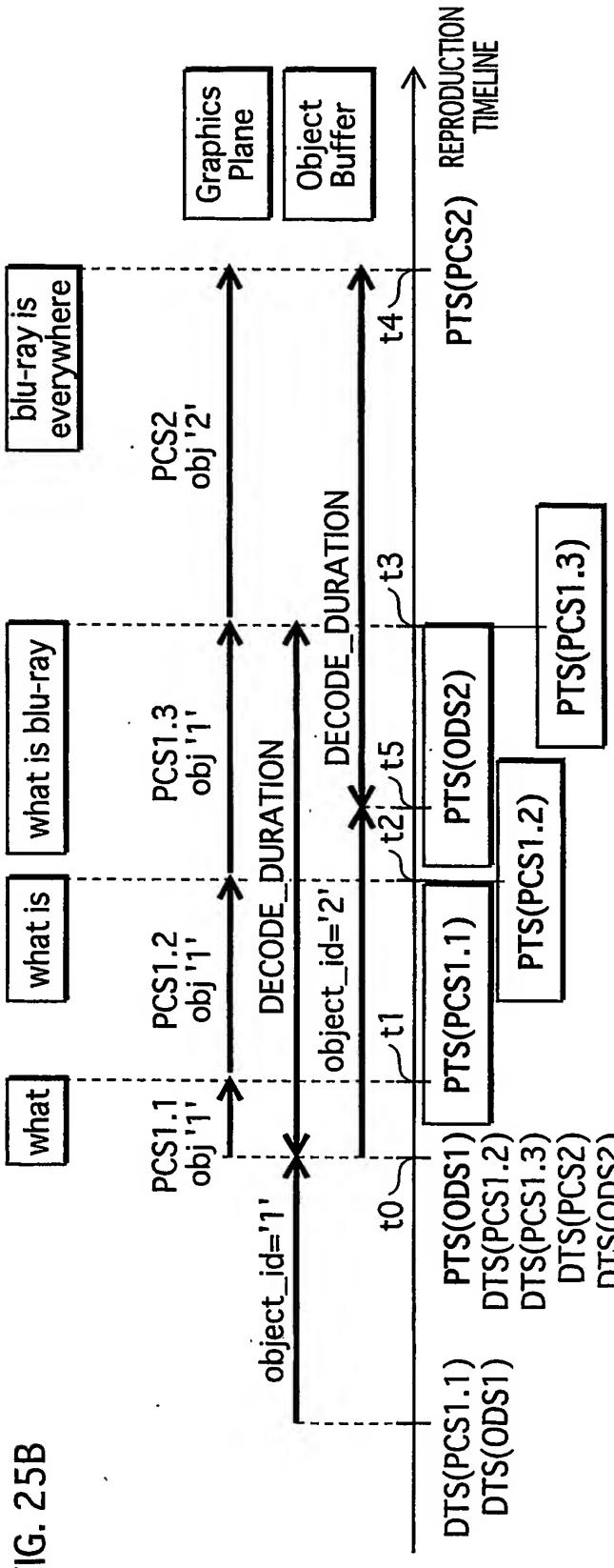
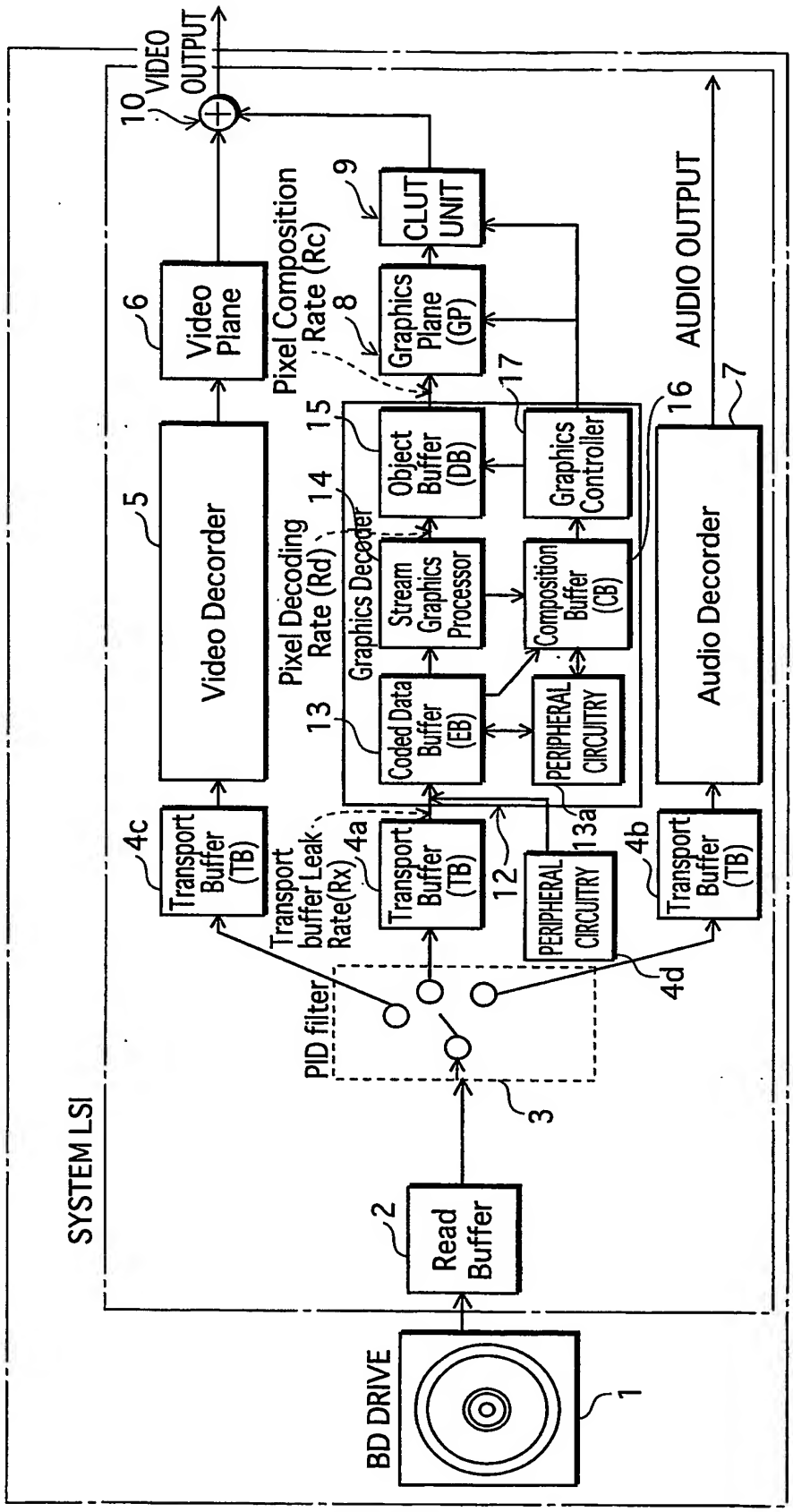


FIG. 26


BD DRIVE
1
SYSTEM LSI
Read Buffer
2
PID filter
3
Transport Buffer (TB)
4a
Transport Buffer (TB)
4b
Transport Buffer (TB)
4c
PERIPHERAL CIRCUITRY
4d
Video Decorder
5
Audio Decorder
7
Video Plane
6
Graphics Plane (GP)
8
CLUT UNIT
9
VIDEO OUTPUT
10
AUDIO OUTPUT
Transport buffer Leak Rate(Rx)
Pixel Decoding Rate (Rd)
Pixel Composition Rate (Rc)
Graphics Decoder
12
PERIPHERAL CIRCUITRY
13a
Coded Data Buffer (EB)
13
Stream Graphics Processor
14
Object Buffer (DB)
15
Composition Buffer (CB)
16
Graphics Controller
17

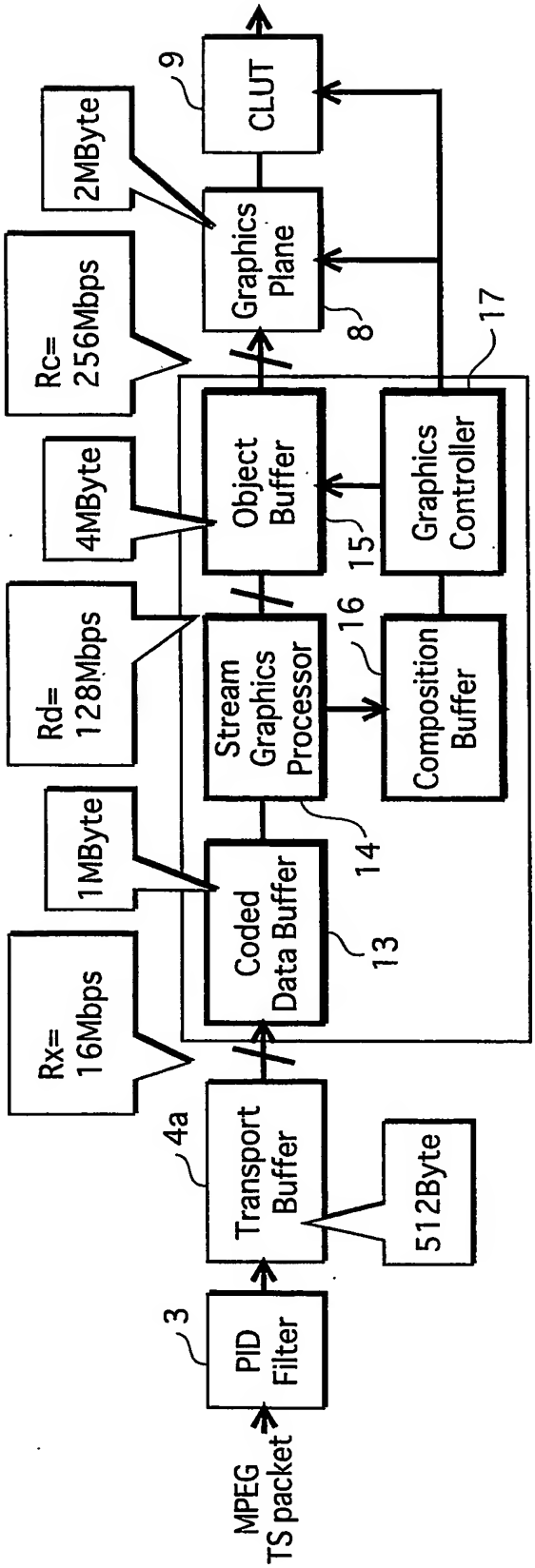

MPEG TS packet
PID Filter
3
Transport Buffer
4a
512Byte
Rx= 16Mbps
Coded Data Buffer
13
1MByte
Rd= 128Mbps
Stream Graphics Processor
14
16
Composition Buffer
4MByte
Object Buffer
15
Graphics Controller
17
Rc= 256Mbps
Graphics Plane
8
2MByte
CLUT
9

FIG. 27

FIG. 28

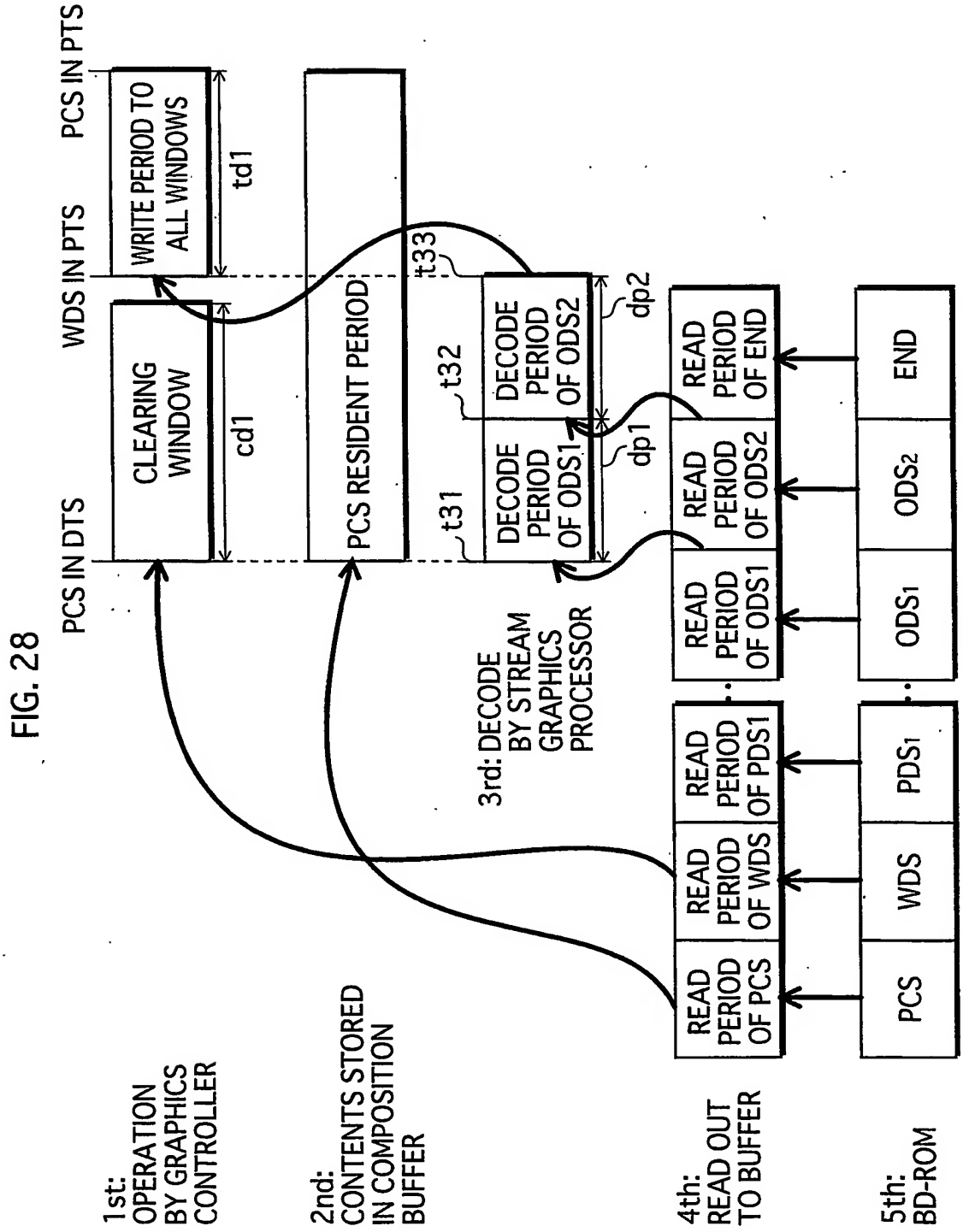
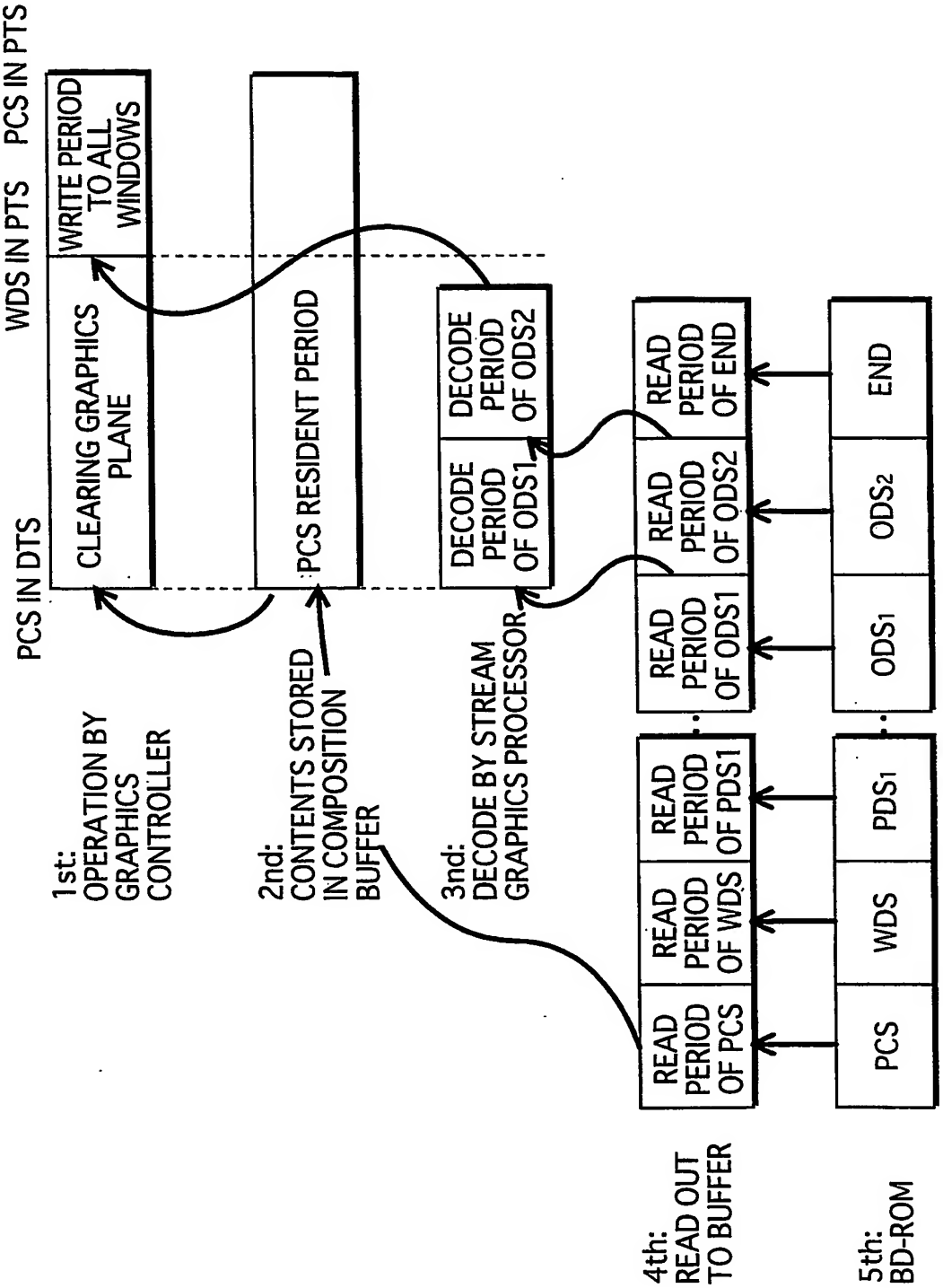

PCS IN DTS
WDS IN PTS
PCS IN PTS
CLEARING WINDOW
cd1
WRITE PERIOD TO ALL WINDOWS
td1
PCS RESIDENT PERIOD
t31
t32
t33
DECODE PERIOD OF ODS1
DECODE PERIOD OF ODS2
dp1
dp2
3rd: DECODE BY STREAM GRAPHICS PROCESSOR
READ PERIOD OF PCS
READ PERIOD OF WDS
READ PERIOD OF PDS1
READ PERIOD OF ODS1
READ PERIOD OF ODS2
READ PERIOD OF END
PCS
WDS
PDS1
ODS1
ODS2
END
1st: OPERATION BY GRAPHICS CONTROLLER
2nd: CONTENTS STORED IN COMPOSITION BUFFER
4th: READ OUT TO BUFFER
5th: BD-ROM

FIG. 29


PCS IN DTS
WDS IN PTS
PCS IN PTS
CLEARING GRAPHICS PLANE
WRITE PERIOD TO ALL WINDOWS
PCS RESIDENT PERIOD
DECODE PERIOD OF ODS1
DECODE PERIOD OF ODS2
1st: OPERATION BY GRAPHICS CONTROLLER
2nd: CONTENTS STORED IN COMPOSITION BUFFER
3nd: DECODE BY STREAM GRAPHICS PROCESSOR
READ PERIOD OF PCS
READ PERIOD OF WDS
READ PERIOD OF PDS1
READ PERIOD OF ODS1
READ PERIOD OF ODS2
READ PERIOD OF END
4th: READ OUT TO BUFFER
PCS
WDS
PDS1
ODS1
ODS2
END
5th: BD-ROM

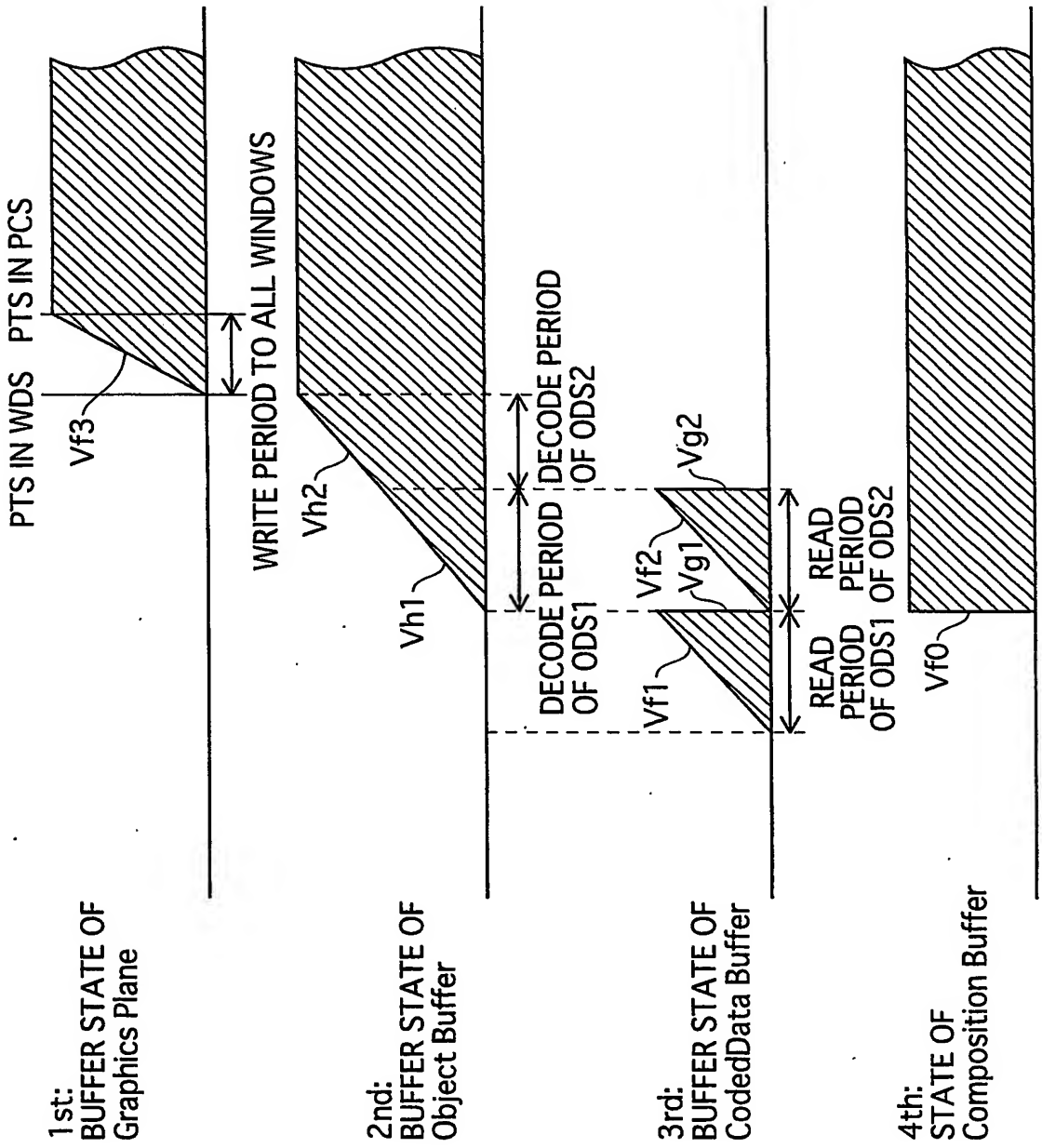

PTS IN WDS
PTS IN PCS
Vf3
WRITE PERIOD TO ALL WINDOWS
1st:
BUFFER STATE OF
Graphics Plane
Vh2
Vh1
DECODE PERIOD
OF ODS1
DECODE PERIOD
OF ODS2
2nd:
BUFFER STATE OF
Object Buffer
Vf2
Vg1
Vg2
Vf1
READ
PERIOD
OF ODS1
READ
PERIOD
OF ODS2
3rd:
BUFFER STATE OF
CodedData Buffer
Vf0
DTS(PCS)
4th:
STATE OF
Composition Buffer

FIG. 30

FIG. 31

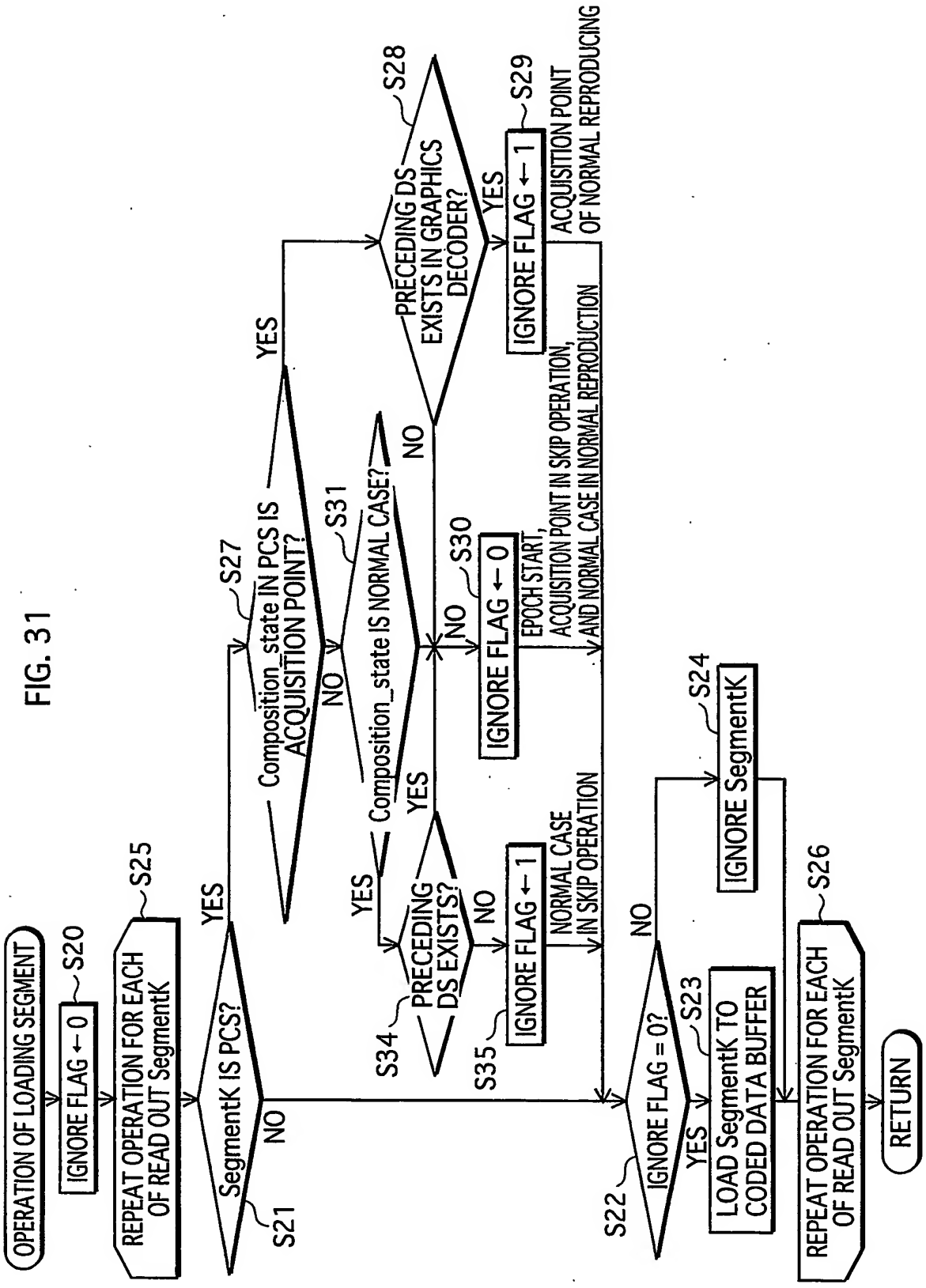

OPERATION OF LOADING SEGMENT
IGNORE FLAG ← 0
S20
REPEAT OPERATION FOR EACH OF READ OUT SegmentK
S25
SegmentK IS PCS?
YES
NO
S21
Composition_state IN PCS IS ACQUISITION POINT?
S27
YES
NO
Composition_state IS NORMAL CASE?
S31
YES
NO
PRECEDING DS EXISTS?
S34
YES
NO
PRECEDING DS EXISTS IN GRAPHICS DECODER?
S28
YES
NO
IGNORE FLAG ← 1
NORMAL CASE IN SKIP OPERATION
S35
IGNORE FLAG ← 0
EPOCH START, ACQUISITION POINT IN SKIP OPERATION, AND NORMAL CASE IN NORMAL REPRODUCTION
S30
IGNORE FLAG ← 1
ACQUISITION POINT OF NORMAL REPRODUCING
S29
IGNORE FLAG = 0?
S22
YES
NO
LOAD SegmentK TO CODED DATA BUFFER
S23
IGNORE SegmentK
S24
REPEAT OPERATION FOR EACH OF READ OUT SegmentK
S26
RETURN

FIG. 32

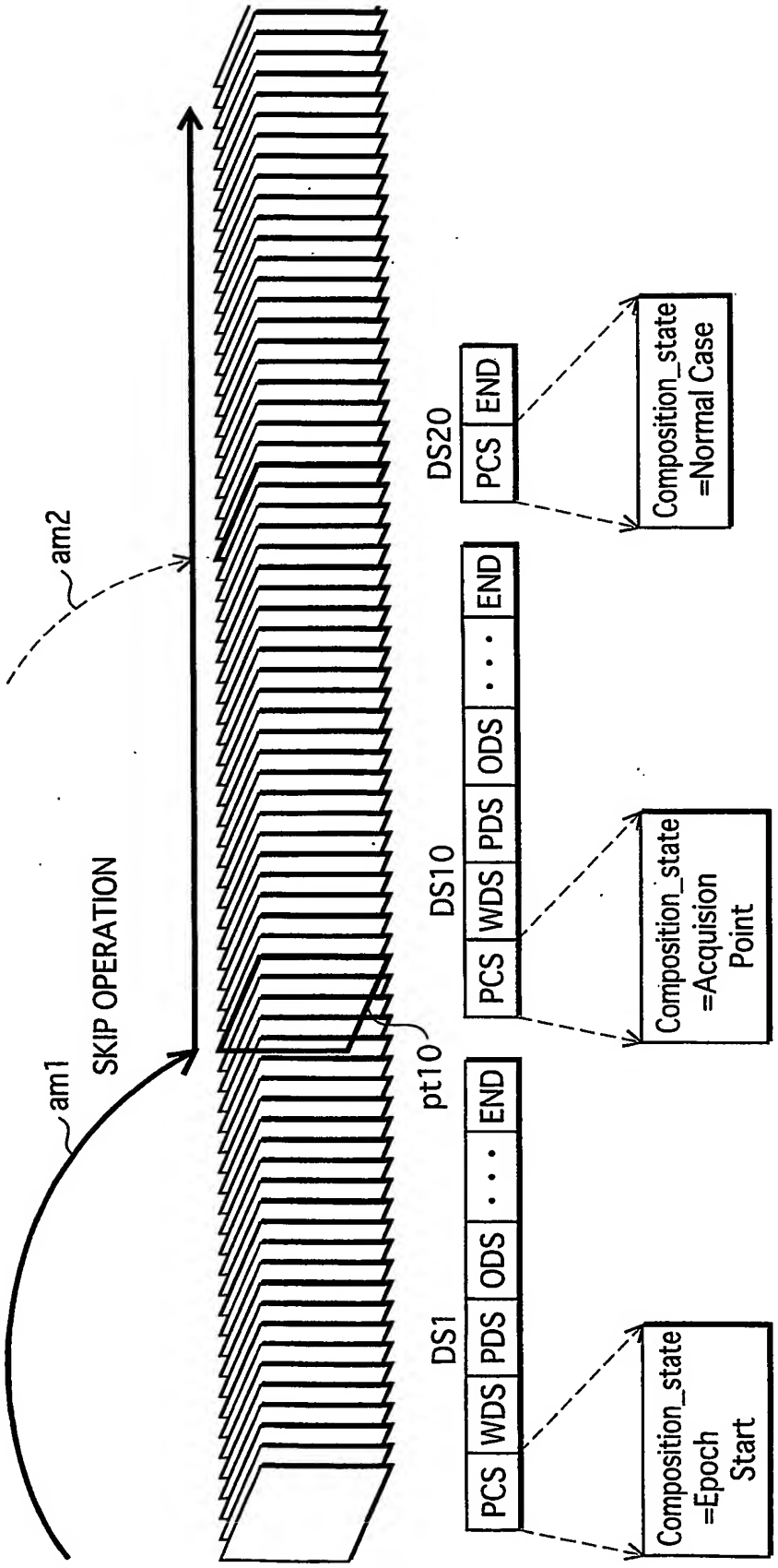

am1
SKIP OPERATION
am2
pt10
DS1
PCS
WDS
PDS
ODS
• • •
END
Composition_state
=Epoch
Start
DS10
PCS
WDS
PDS
ODS
• • •
END
Composition_state
=Acquision
Point
DS20
PCS
END
Composition_state
=Normal Case

FIG. 33

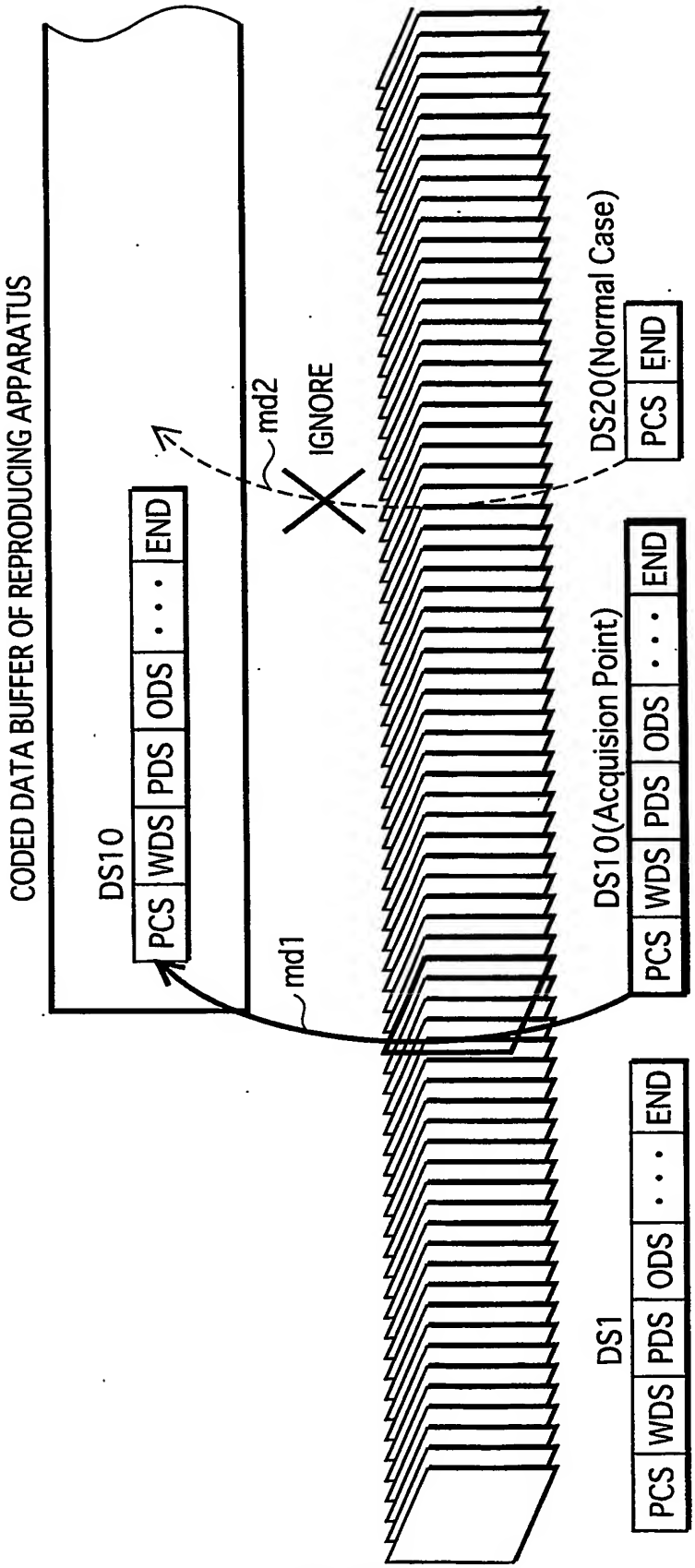

CODED DATA BUFFER OF REPRODUCING APPARATUS
DS10
PCS
WDS
PDS
ODS
• • •
END
md1
md2
IGNORE
DS20(Normal Case)
PCS
END
DS10(Acquision Point)
PCS
WDS
PDS
ODS
• • •
END
DS1
PCS
WDS
PDS
ODS
• • •
END

FIG. 34

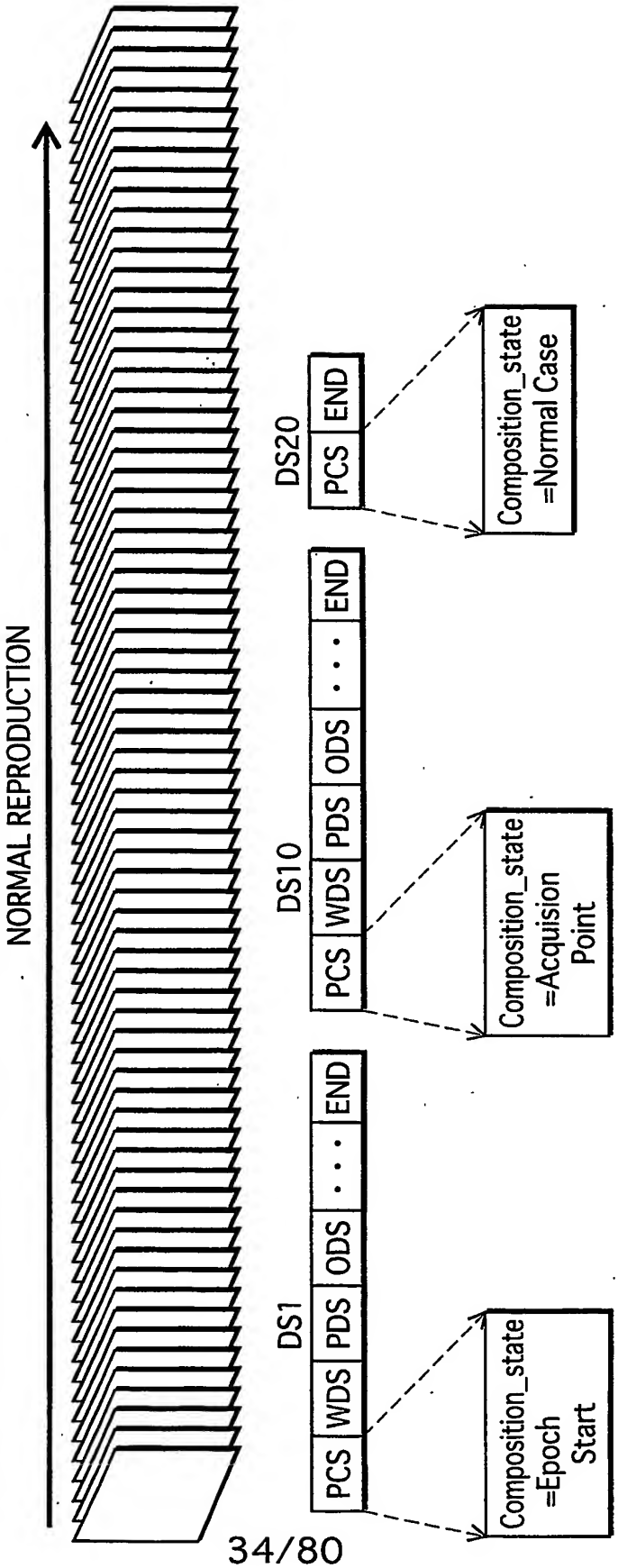

NORMAL REPRODUCTION
DS1
PCS
WDS
PDS
ODS
• • •
END
Composition_state
=Epoch
Start
DS10
PCS
WDS
PDS
ODS
• • •
END
Composition_state
=Acquision
Point
DS20
PCS
END
Composition_state
=Normal Case

FIG. 35

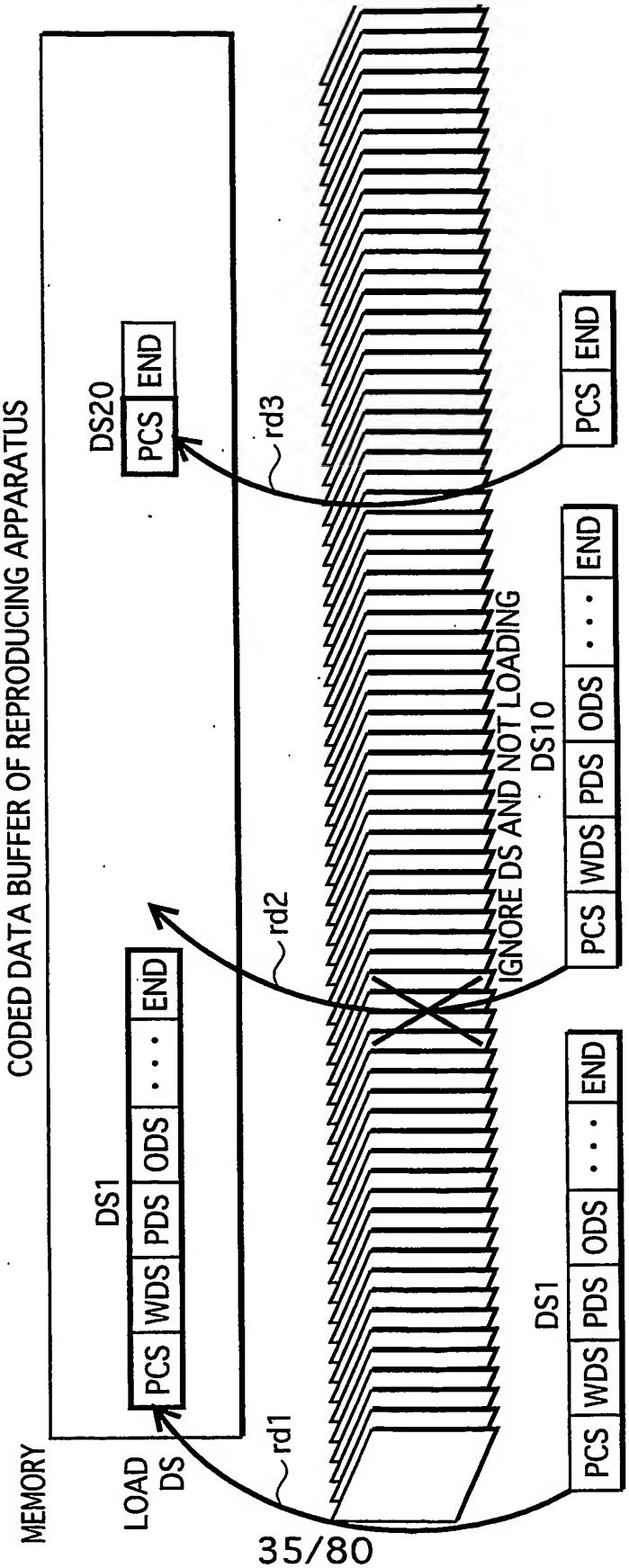

MEMORY
CODED DATA BUFFER OF REPRODUCING APPARATUS
LOAD DS
rd1
rd2
rd3
DS1
PCS
WDS
PDS
ODS
• • •
END
DS20
IGNORE DS AND NOT LOADING
DS10

FIG. 36

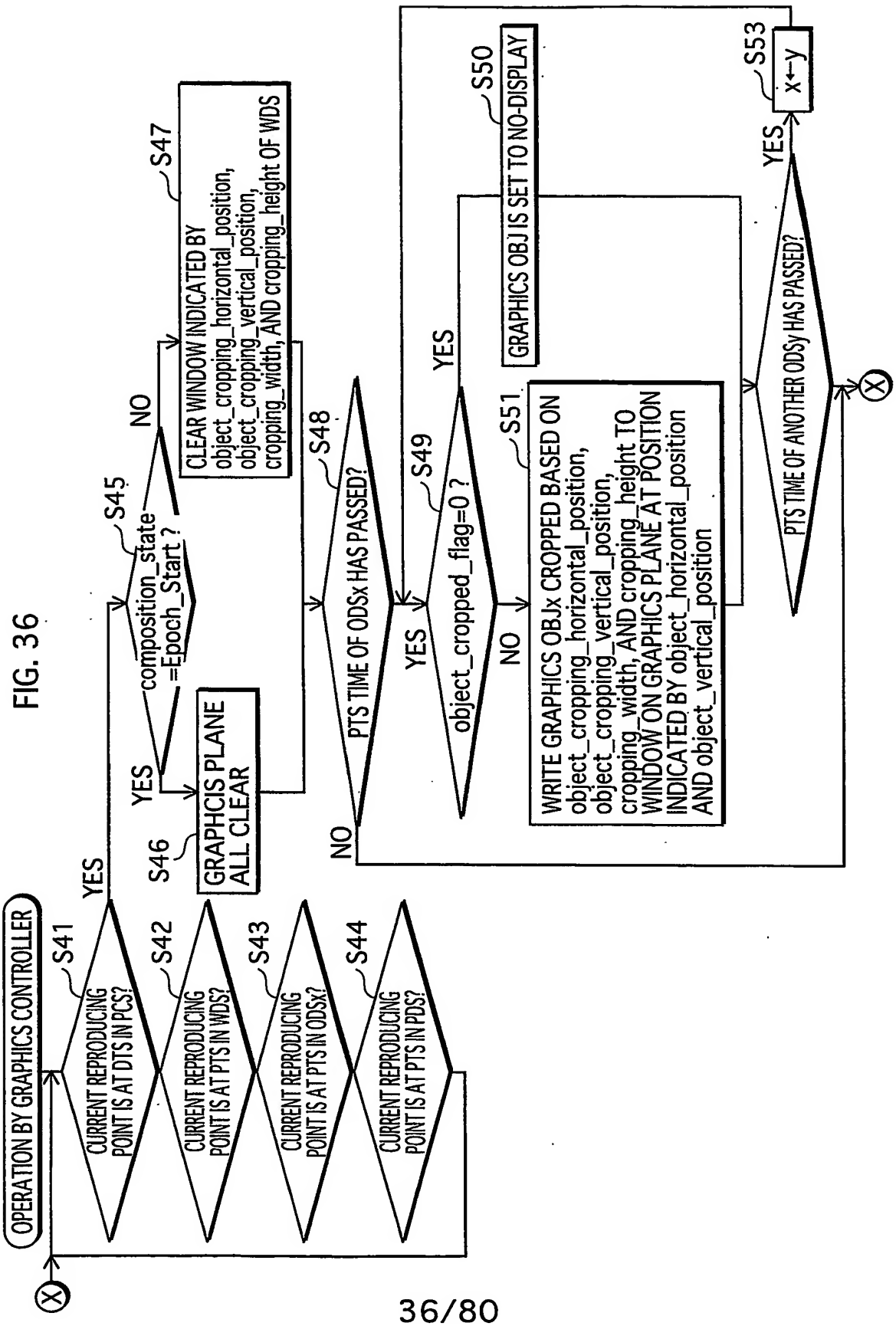

OPERATION BY GRAPHICS CONTROLLER
S41
CURRENT REPRODUCING POINT IS AT DTS IN PCS?
YES
S42
CURRENT REPRODUCING POINT IS AT PTS IN WDS?
YES
S43
CURRENT REPRODUCING POINT IS AT PTS IN ODSx?
YES
S44
CURRENT REPRODUCING POINT IS AT PTS IN PDS?
YES
S45
composition_state =Epoch_Start ?
NO
S46
GRAPHCIS PLANE ALL CLEAR
S47
CLEAR WINDOW INDICATED BY object_cropping_horizontal_position, object_cropping_vertical_position, cropping_width, AND cropping_height OF WDS
S48
PTS TIME OF ODSx HAS PASSED?
NO
S49
object_cropped_flag=0 ?
YES
S50
GRAPHICS OBJ IS SET TO NO-DISPLAY
S51
NO
WRITE GRAPHICS OBJx CROPPED BASED ON object_cropping_horizontal_position, object_cropping_vertical_position, cropping_width, AND cropping_height TO WINDOW ON GRAPHICS PLANE AT POSITION INDICATED BY object_horizontal_position AND object_vertical_position
S53
x←y
YES
PTS TIME OF ANOTHER ODSy HAS PASSED?

FIG. 37

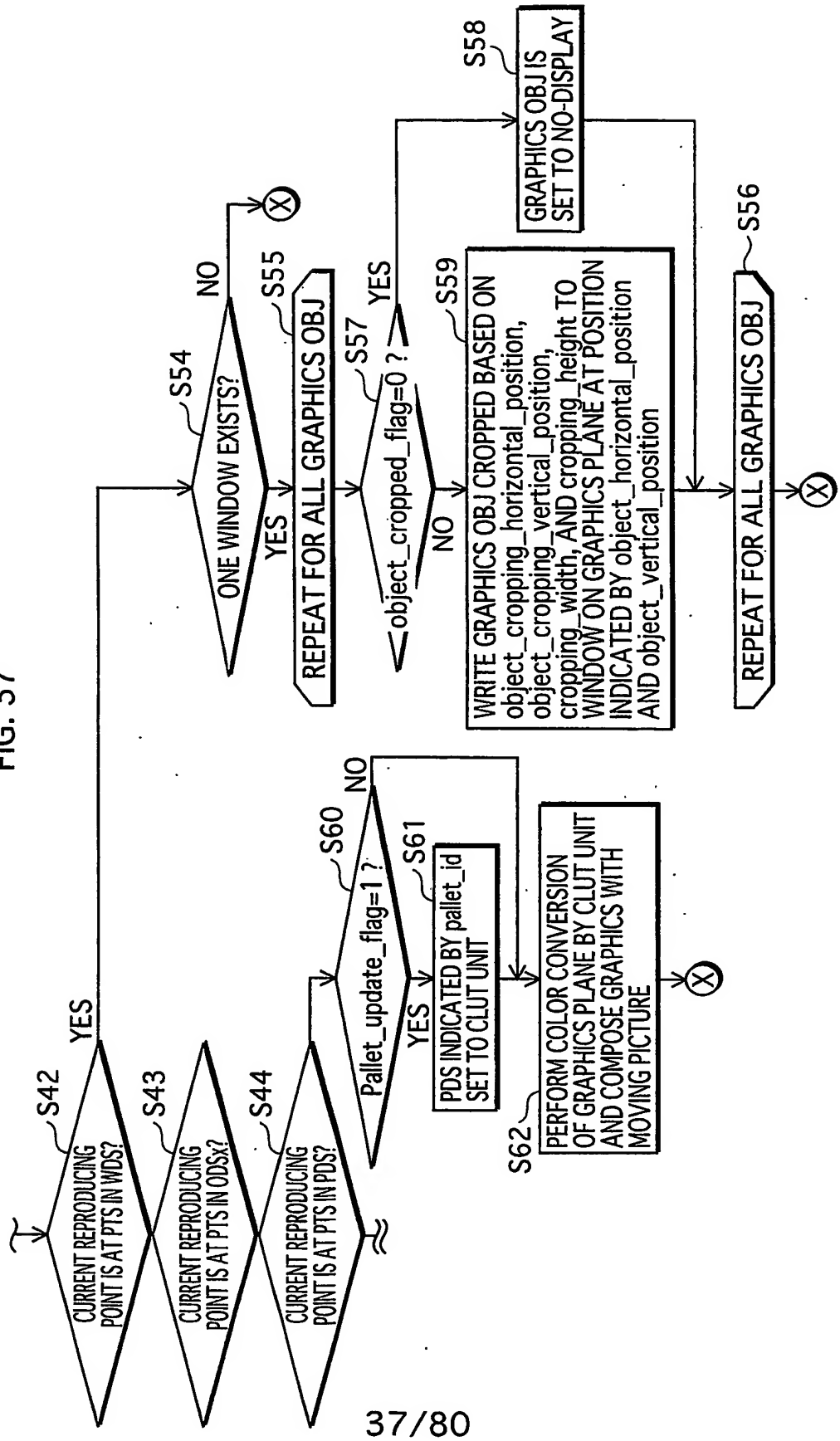

CURRENT REPRODUCING POINT IS AT PTS IN WDS?
S42
YES
CURRENT REPRODUCING POINT IS AT PTS IN ODSx?
S43
CURRENT REPRODUCING POINT IS AT PTS IN PDS?
S44
Pallet_update_flag=1 ?
S60
NO
YES
PDS INDICATED BY pallet_id SET TO CLUT UNIT
S61
PERFORM COLOR CONVERSION OF GRAPHICS PLANE BY CLUT UNIT AND COMPOSE GRAPHICS WITH MOVING PICTURE
S62
ONE WINDOW EXISTS?
S54
NO
YES
REPEAT FOR ALL GRAPHICS OBJ
S55
object_cropped_flag=0 ?
S57
YES
NO
WRITE GRAPHICS OBJ CROPPED BASED ON object_cropping_horizontal_position, object_cropping_vertical_position, cropping_width, AND cropping_height TO WINDOW ON GRAPHICS PLANE AT POSITION INDICATED BY object_horizontal_position AND object_vertical_position
S59
GRAPHICS OBJ IS SET TO NO-DISPLAY
S58
REPEAT FOR ALL GRAPHICS OBJ
S56

FIG. 38

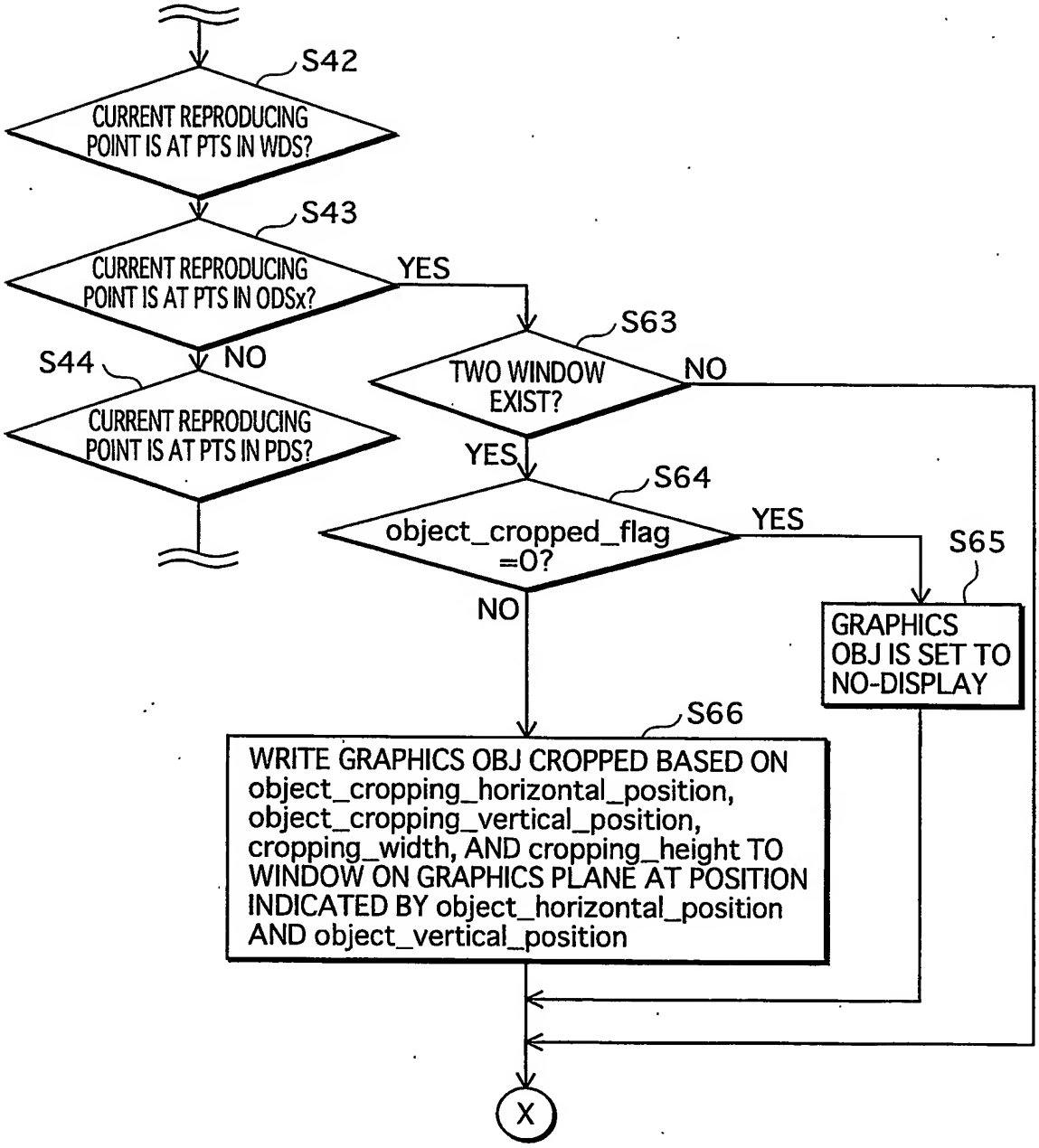

S42
CURRENT REPRODUCING POINT IS AT PTS IN WDS?
S43
CURRENT REPRODUCING POINT IS AT PTS IN ODSx?
YES
NO
S44
CURRENT REPRODUCING POINT IS AT PTS IN PDS?
S63
TWO WINDOW EXIST?
NO
YES
S64
object_cropped_flag =0?
YES
NO
S65
GRAPHICS OBJ IS SET TO NO-DISPLAY
S66
WRITE GRAPHICS OBJ CROPPED BASED ON object_cropping_horizontal_position, object_cropping_vertical_position, cropping_width, AND cropping_height TO WINDOW ON GRAPHICS PLANE AT POSITION INDICATED BY object_horizontal_position AND object_vertical_position
X

FIG. 39

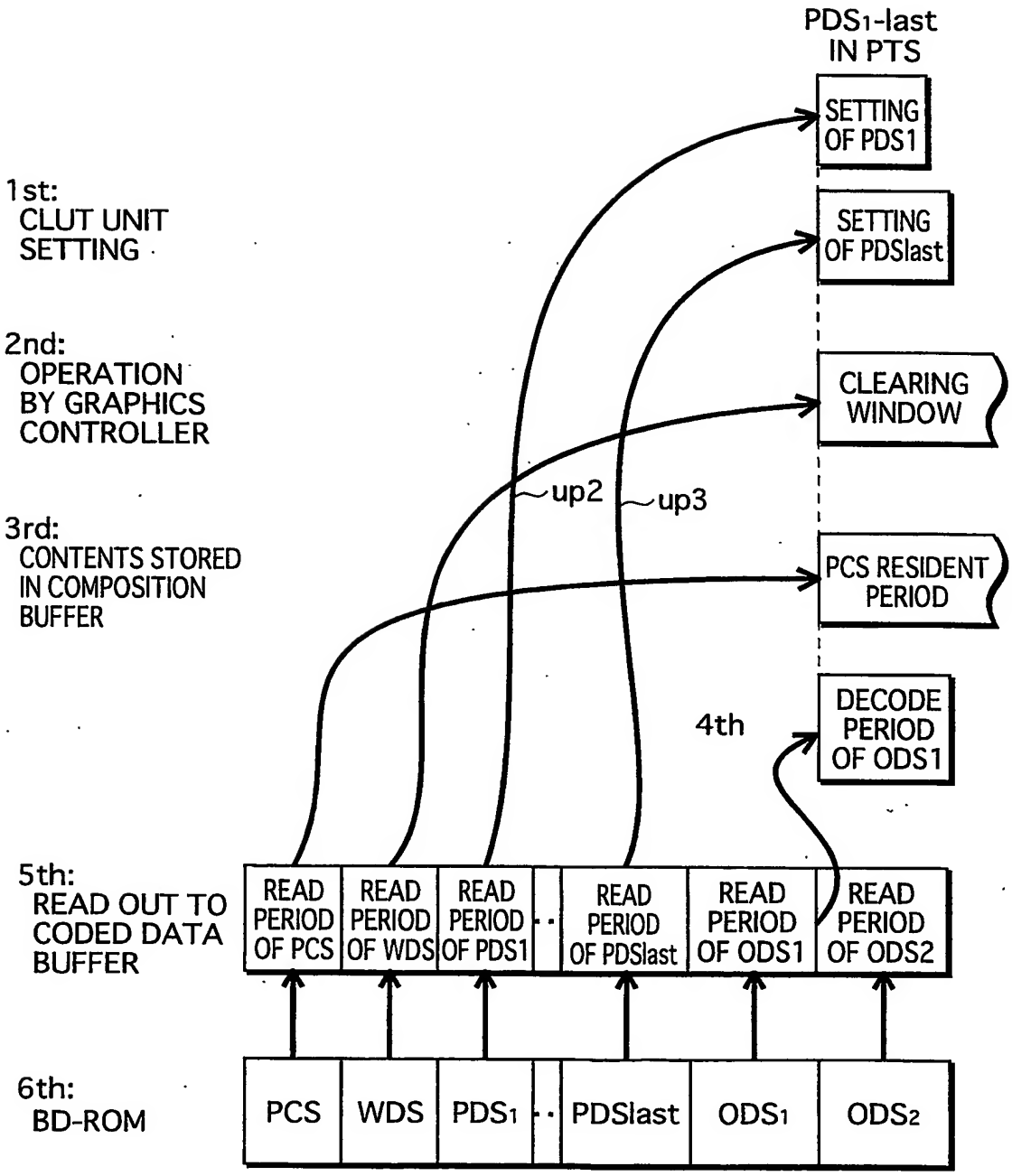

PDS1-last IN PTS
SETTING OF PDS1
SETTING OF PDSlast
CLEARING WINDOW
PCS RESIDENT PERIOD
DECODE PERIOD OF ODS1
1st: CLUT UNIT SETTING
2nd: OPERATION BY GRAPHICS CONTROLLER
3rd: CONTENTS STORED IN COMPOSITION BUFFER
4th
up2
up3
5th: READ OUT TO CODED DATA BUFFER
READ PERIOD OF PCS
READ PERIOD OF WDS
READ PERIOD OF PDS1
READ PERIOD OF PDSlast
READ PERIOD OF ODS1
READ PERIOD OF ODS2
6th: BD-ROM
PCS
WDS
PDS1
PDSlast
ODS1
ODS2

FIG. 40

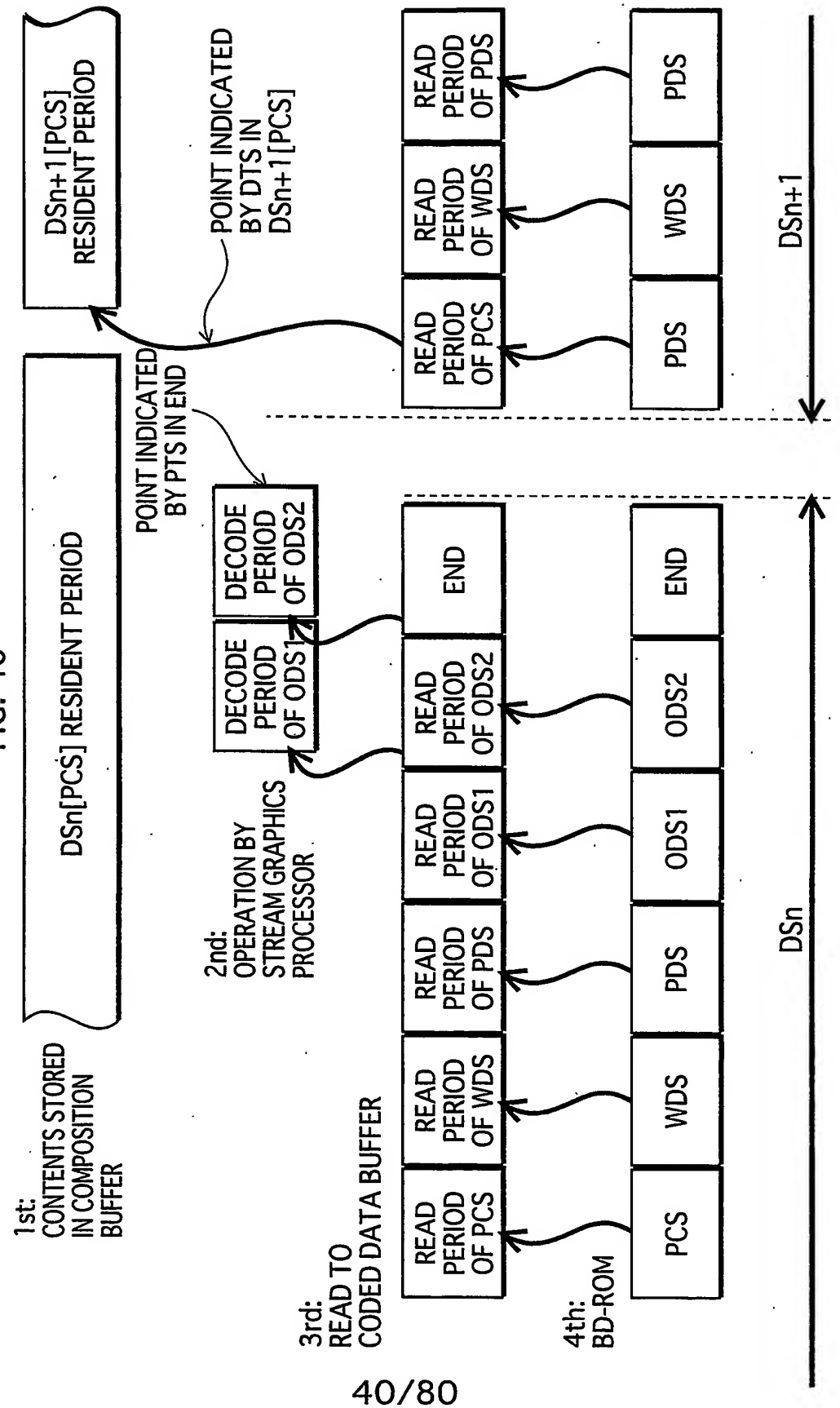

1st:
CONTENTS STORED
IN COMPOSITION
BUFFER
DSn[PCS] RESIDENT PERIOD
DSn+1[PCS]
RESIDENT PERIOD
POINT INDICATED
BY PTS IN END
POINT INDICATED
BY DTS IN
DSn+1[PCS]
2nd:
OPERATION BY
STREAM GRAPHICS
PROCESSOR
DECODE
PERIOD
OF ODS1
DECODE
PERIOD
OF ODS2
3rd:
READ TO
CODED DATA BUFFER
READ
PERIOD
OF PCS
READ
PERIOD
OF WDS
READ
PERIOD
OF PDS
READ
PERIOD
OF ODS1
READ
PERIOD
OF ODS2
END
4th:
BD-ROM
PCS
WDS
PDS
ODS1
ODS2
END
DSn
READ
PERIOD
OF PCS
READ
PERIOD
OF WDS
READ
PERIOD
OF PDS
PDS
WDS
PDS
DSn+1

FIG. 41

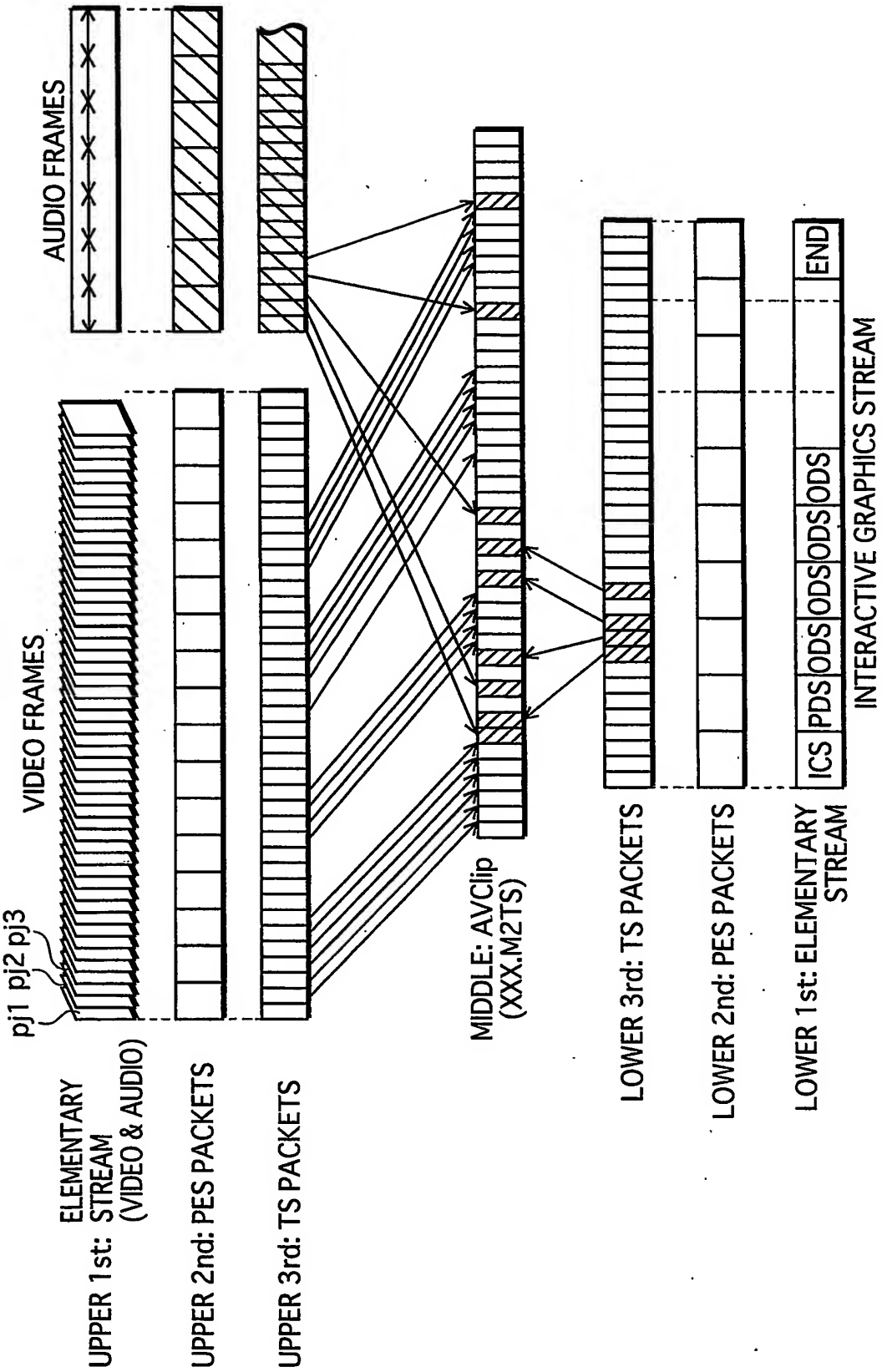

AUDIO FRAMES
VIDEO FRAMES
pj1 pj2 pj3
UPPER 1st: ELEMENTARY STREAM (VIDEO & AUDIO)
UPPER 2nd: PES PACKETS
UPPER 3rd: TS PACKETS
MIDDLE: AVClip (XXX.M2TS)
LOWER 3rd: TS PACKETS
LOWER 2nd: PES PACKETS
LOWER 1st: ELEMENTARY STREAM
ICS
PDS
ODS
ODS
ODS
ODS
END
INTERACTIVE GRAPHICS STREAM

FIG.42A

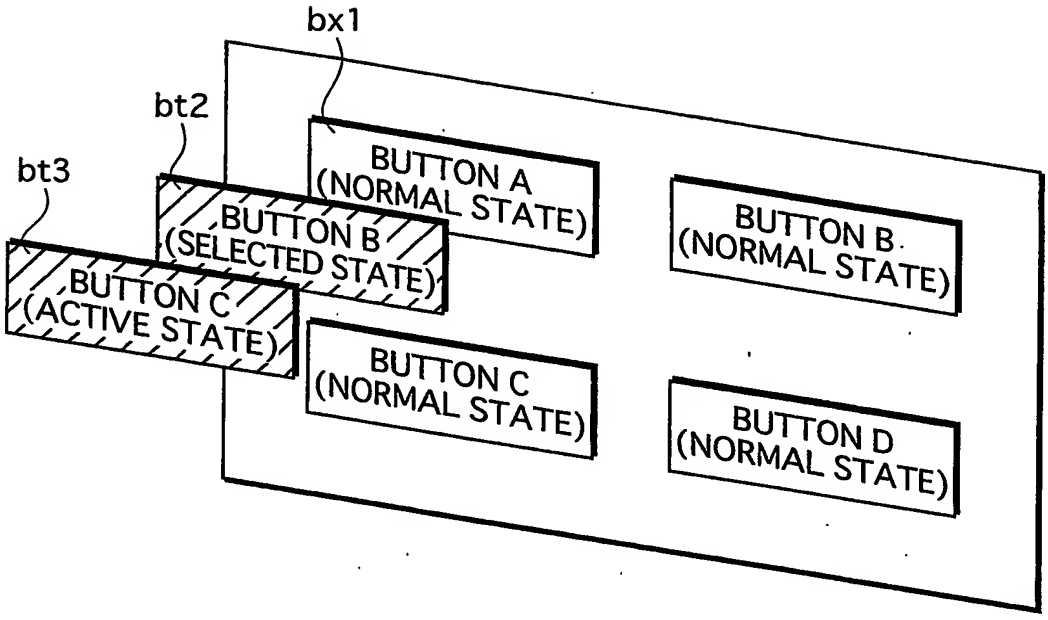


FIG.42B

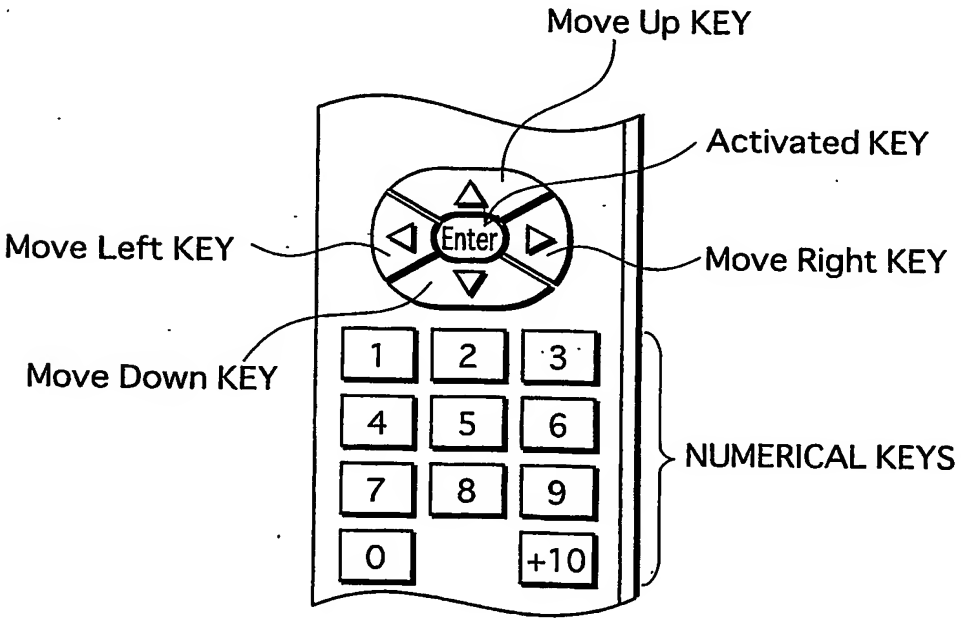


FIG.43

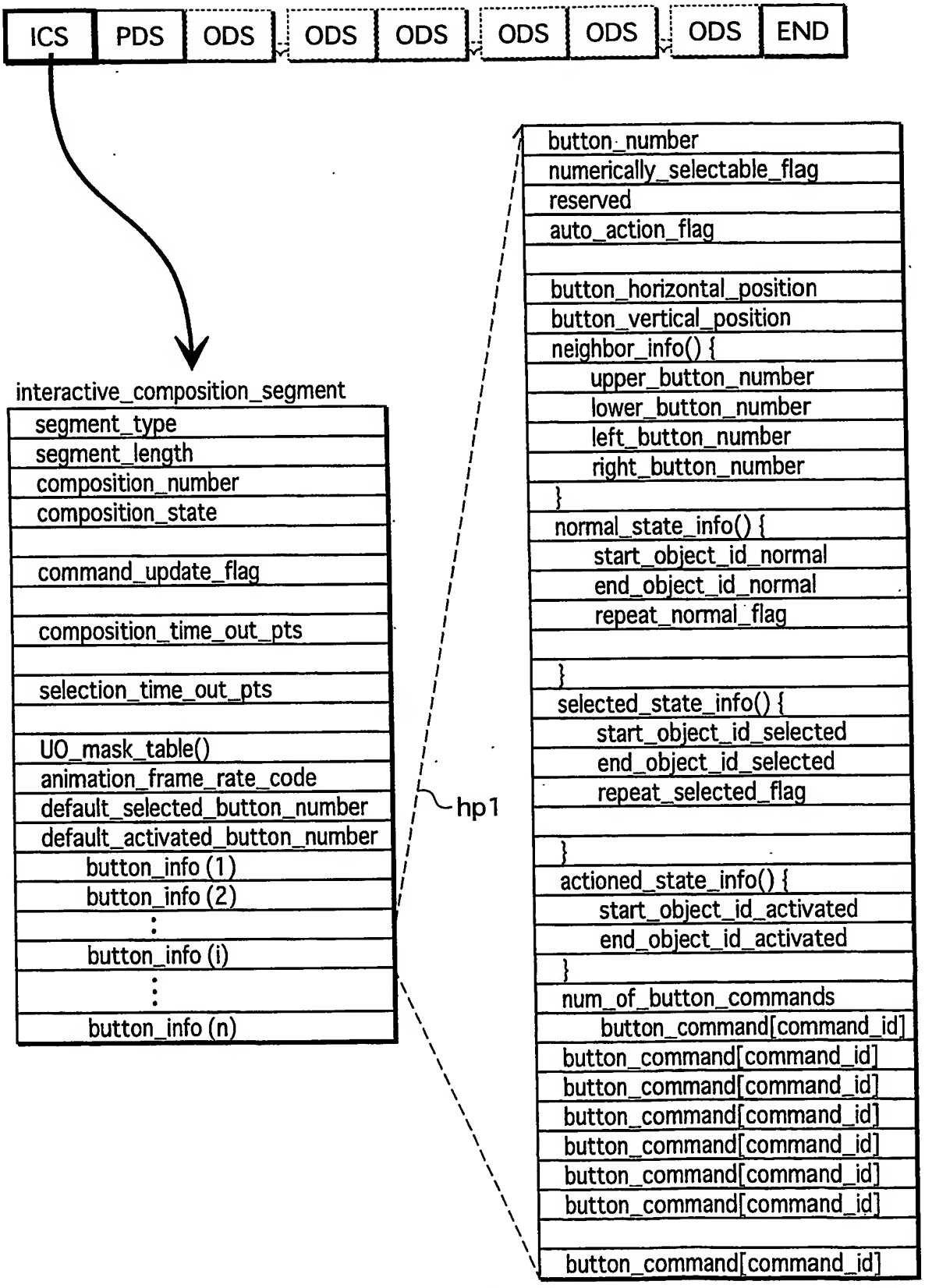

ICS
PDS
ODS
ODS
ODS
ODS
ODS
ODS
END
interactive_composition_segment
segment_type
segment_length
composition_number
composition_state
command_update_flag
composition_time_out_pts
selection_time_out_pts
UO_mask_table()
animation_frame_rate_code
default_selected_button_number
default_activated_button_number
button_info (1)
button_info (2)
button_info (i)
button_info (n)
hp1
button_number
numerically_selectable_flag
reserved
auto_action_flag
button_horizontal_position
button_vertical_position
neighbor_info() {
upper_button_number
lower_button_number
left_button_number
right_button_number
}
normal_state_info() {
start_object_id_normal
end_object_id_normal
repeat_normal_flag
}
selected_state_info() {
start_object_id_selected
end_object_id_selected
repeat_selected_flag
}
actioned_state_info() {
start_object_id_activated
end_object_id_activated
}
num_of_button_commands
button_command[command_id]
button_command[command_id]
button_command[command_id]
button_command[command_id]
button_command[command_id]
button_command[command_id]
button_command[command_id]
button_command[command_id]

FIG. 44

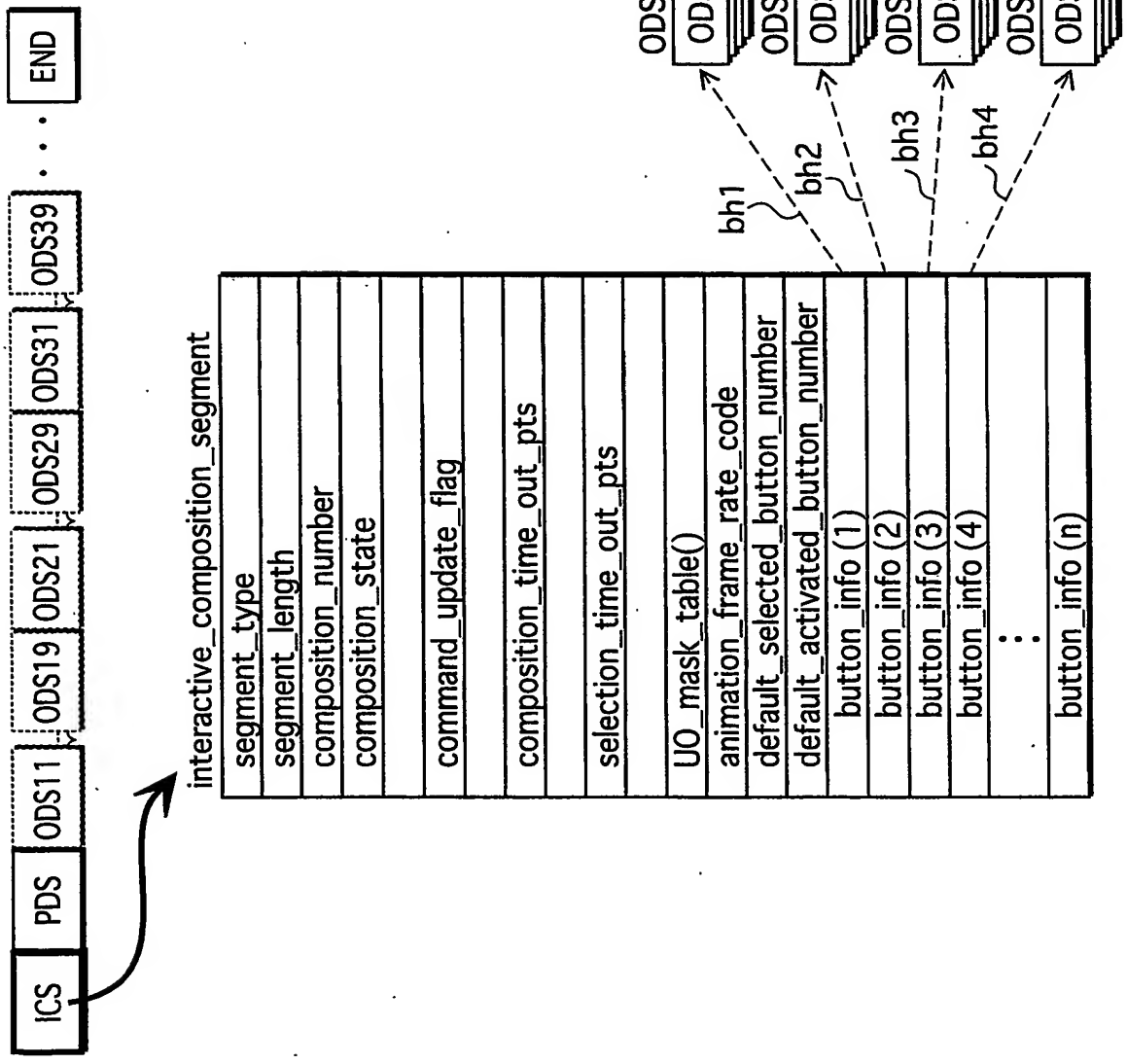

ICS
PDS
ODS11
ODS19
ODS21
ODS29
ODS31
ODS39
END
interactive_composition_segment
segment_type
segment_length
composition_number
composition_state
command_update_flag
composition_time_out_pts
selection_time_out_pts
UO_mask_table()
animation_frame_rate_code
default_selected_button_number
default_activated_button_number
button_info (1)
button_info (2)
button_info (3)
button_info (4)
button_info (n)
bh1
bh2
bh3
bh4
BUTTON
BUTTON A
BUTTON B
BUTTON C
BUTTON D
ODS11-19
ODS11
ODS21-29
ODS21
ODS31-39
ODS31
ODS41-49
ODS41

FIG. 45

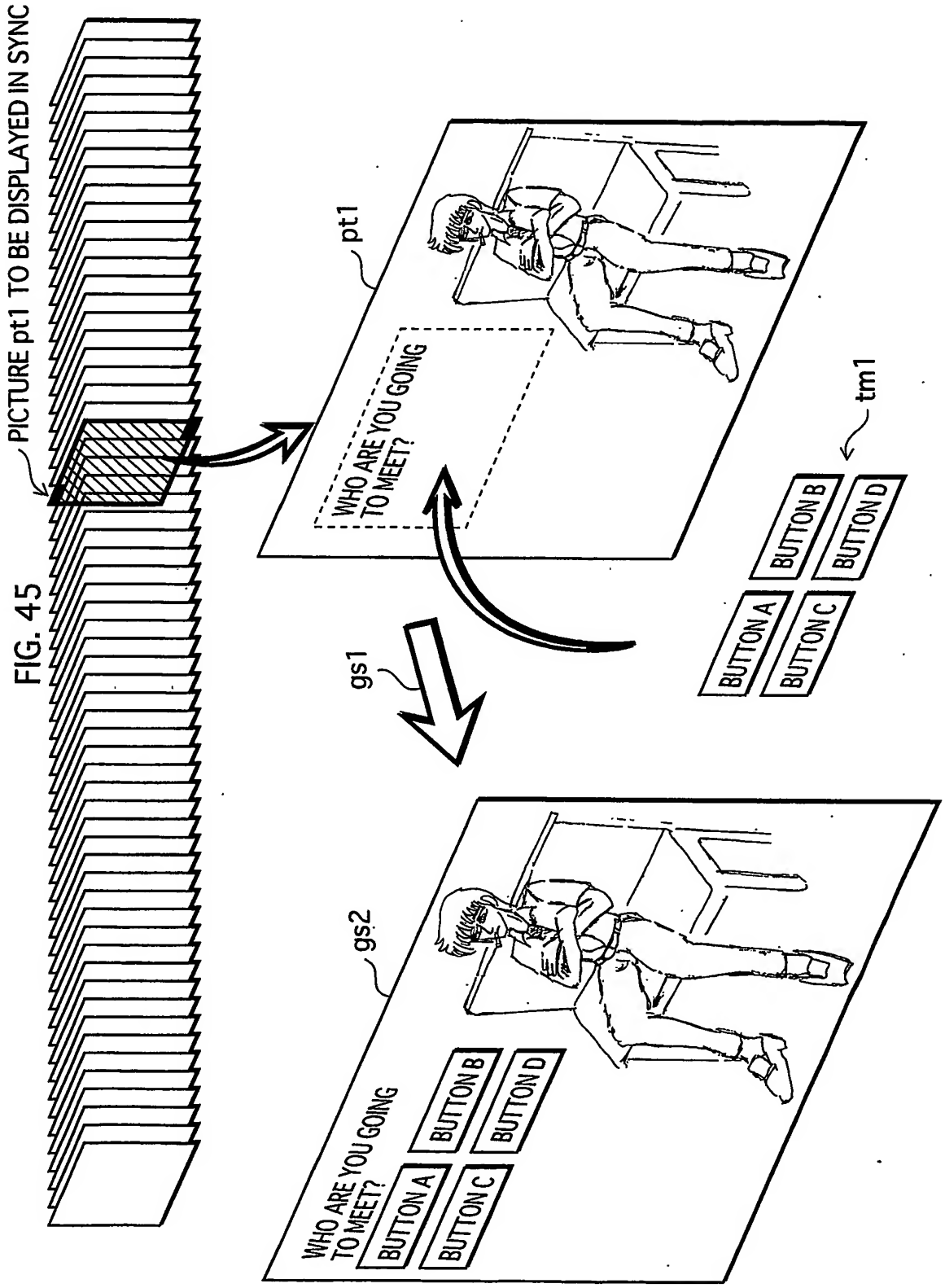
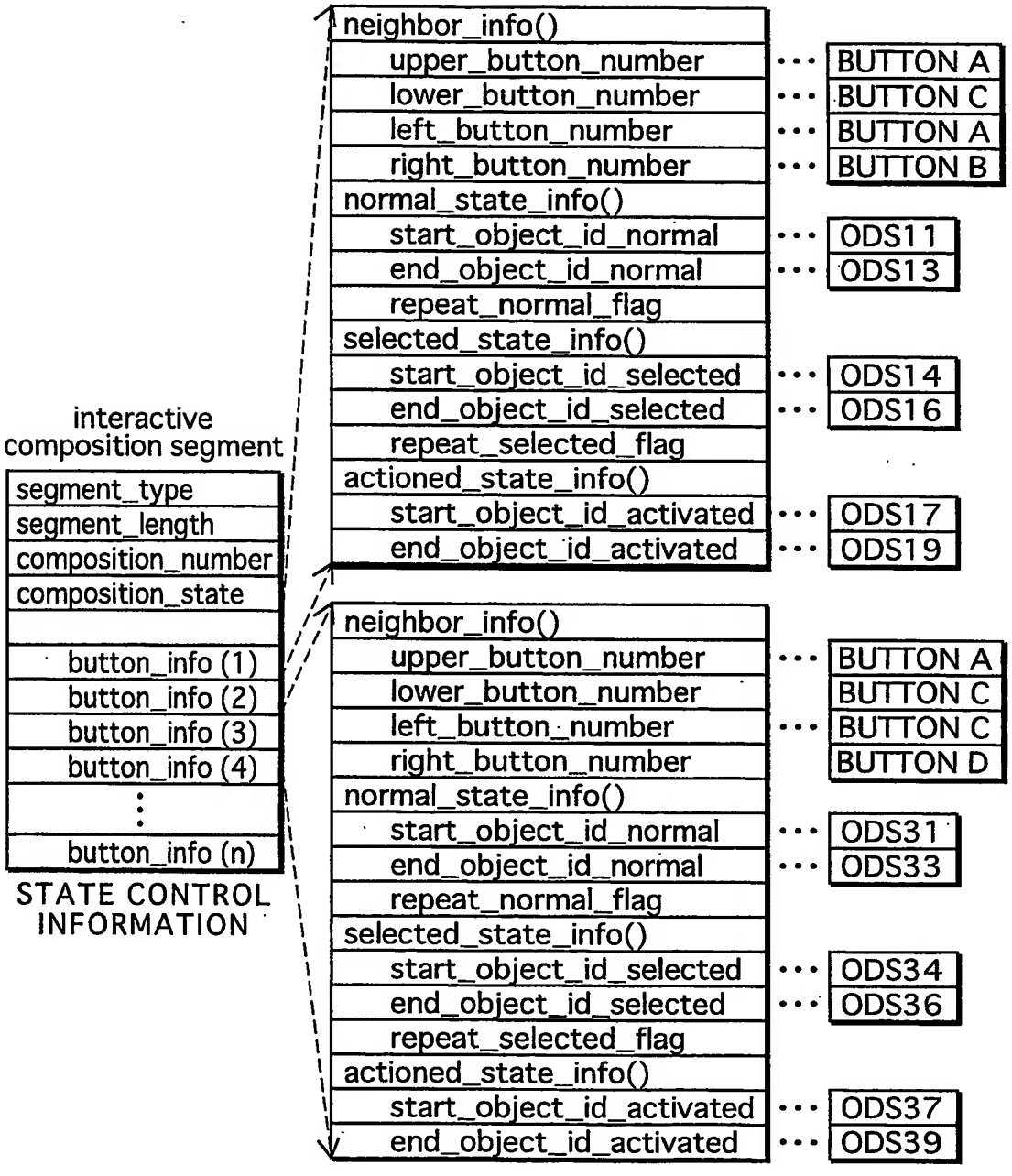

PICTURE pt1 TO BE DISPLAYED IN SYNC
pt1
WHO ARE YOU GOING TO MEET?
tm1
BUTTON B
BUTTON D
BUTTON A
BUTTON C
gs1
gs2
WHO ARE YOU GOING TO MEET?
BUTTON B
BUTTON D
BUTTON A
BUTTON C

FIG.46


interactive composition segment
segment_type
segment_length
composition_number
composition_state
button_info (1)
button_info (2)
button_info (3)
button_info (4)
button_info (n)
STATE CONTROL INFORMATION
neighbor_info()
upper_button_number
lower_button_number
left_button_number
right_button_number
normal_state_info()
start_object_id_normal
end_object_id_normal
repeat_normal_flag
selected_state_info()
start_object_id_selected
end_object_id_selected
repeat_selected_flag
actioned_state_info()
start_object_id_activated
end_object_id_activated
BUTTON A
BUTTON C
BUTTON A
BUTTON B
ODS11
ODS13
ODS14
ODS16
ODS17
ODS19
neighbor_info()
upper_button_number
lower_button_number
left_button_number
right_button_number
normal_state_info()
start_object_id_normal
end_object_id_normal
repeat_normal_flag
selected_state_info()
start_object_id_selected
end_object_id_selected
repeat_selected_flag
actioned_state_info()
start_object_id_activated
end_object_id_activated
BUTTON A
BUTTON C
BUTTON C
BUTTON D
ODS31
ODS33
ODS34
ODS36
ODS37
ODS39

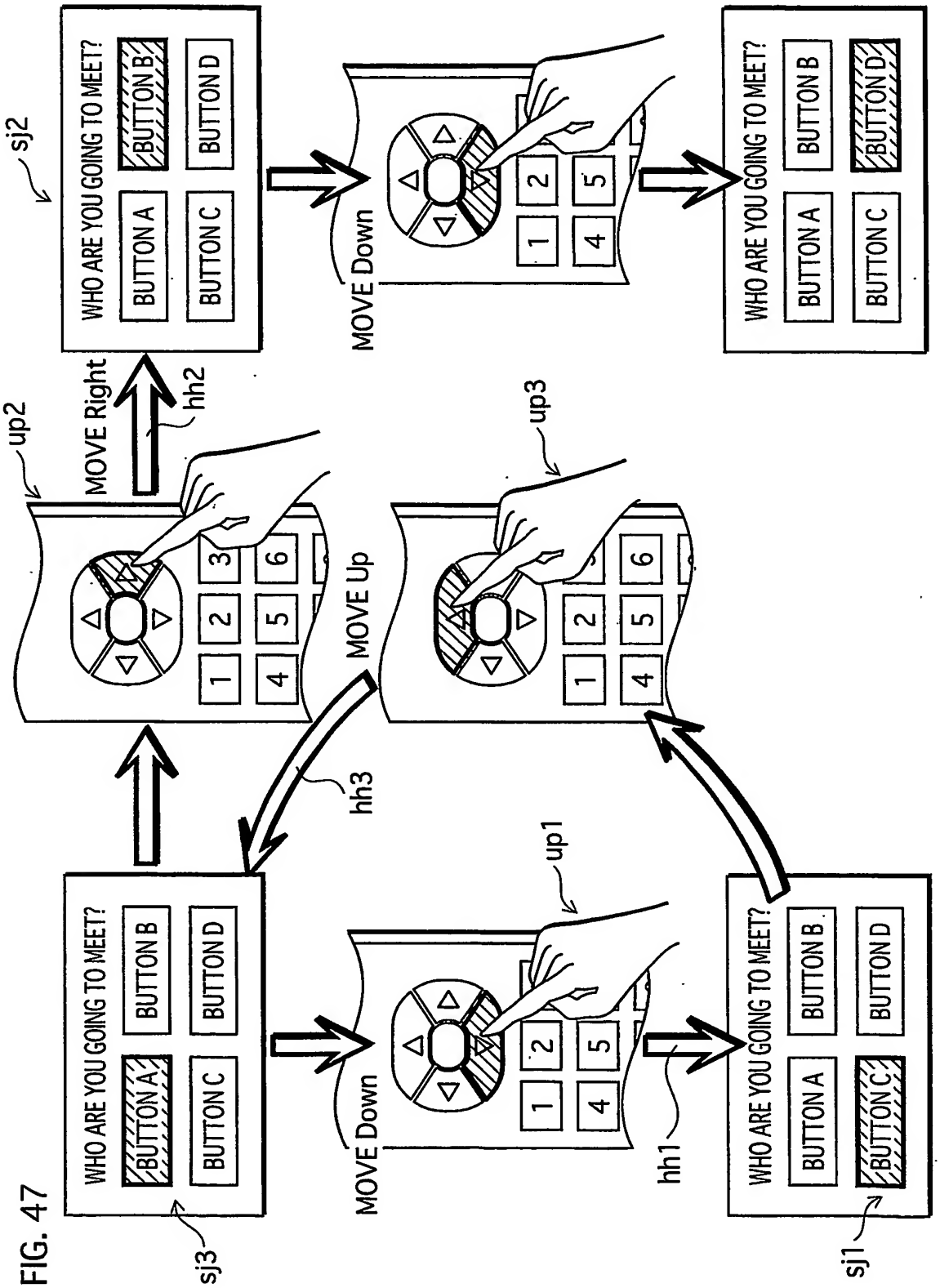

WHO ARE YOU GOING TO MEET?
BUTTON A
BUTTON B
BUTTON C
BUTTON D
sj2
sj3
sj1
up2
up3
up1
MOVE Right
MOVE Down
MOVE Up
hh2
hh3
hh1
1
2
3
4
5
6

FIG. 47

FIG.48

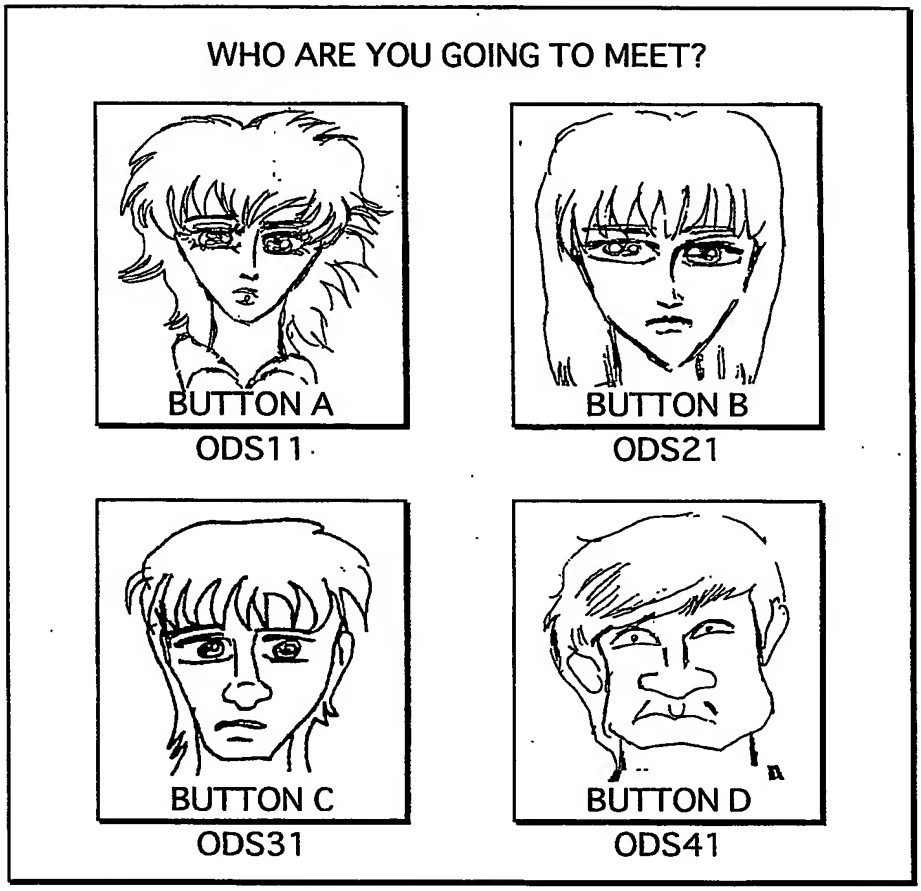

WHO ARE YOU GOING TO MEET?
BUTTON A
ODS11
BUTTON B
ODS21
BUTTON C
ODS31
BUTTON D
ODS41

FIG.49

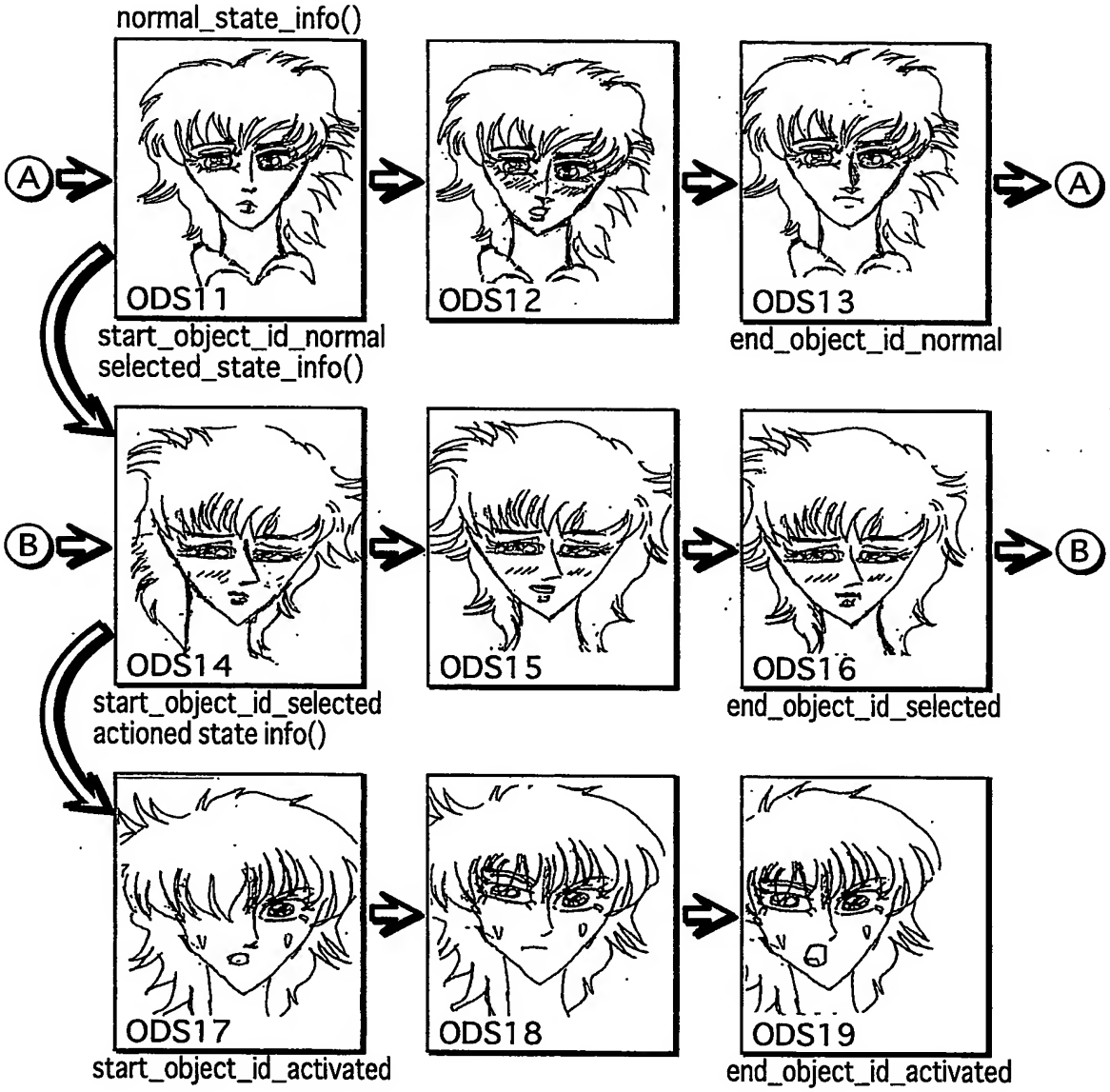

normal_state_info()
A
ODS11
ODS12
ODS13
A
start_object_id_normal
selected_state_info()
end_object_id_normal
B
ODS14
ODS15
ODS16
B
start_object_id_selected
actioned state info()
end_object_id_selected
ODS17
ODS18
ODS19
start_object_id_activated
end_object_id_activated

FIG. 50

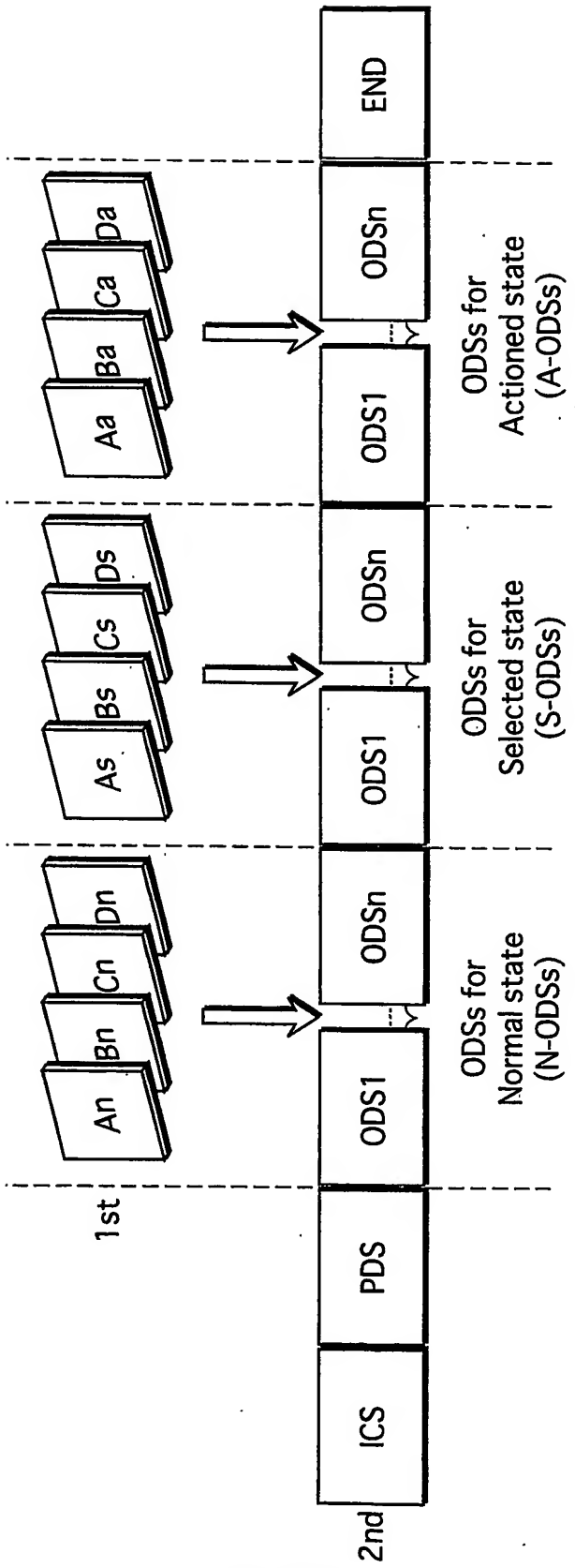

ICS
PDS
ODS1
ODSn
ODS1
ODSn
ODS1
ODSn
END
2nd
1st
An
Bn
Cn
Dn
As
Bs
Cs
Ds
Aa
Ba
Ca
Da
ODSs for Normal state (N-ODSs)
ODSs for Selected state (S-ODSs)
ODSs for Actioned state (A-ODSs)

FIG.51

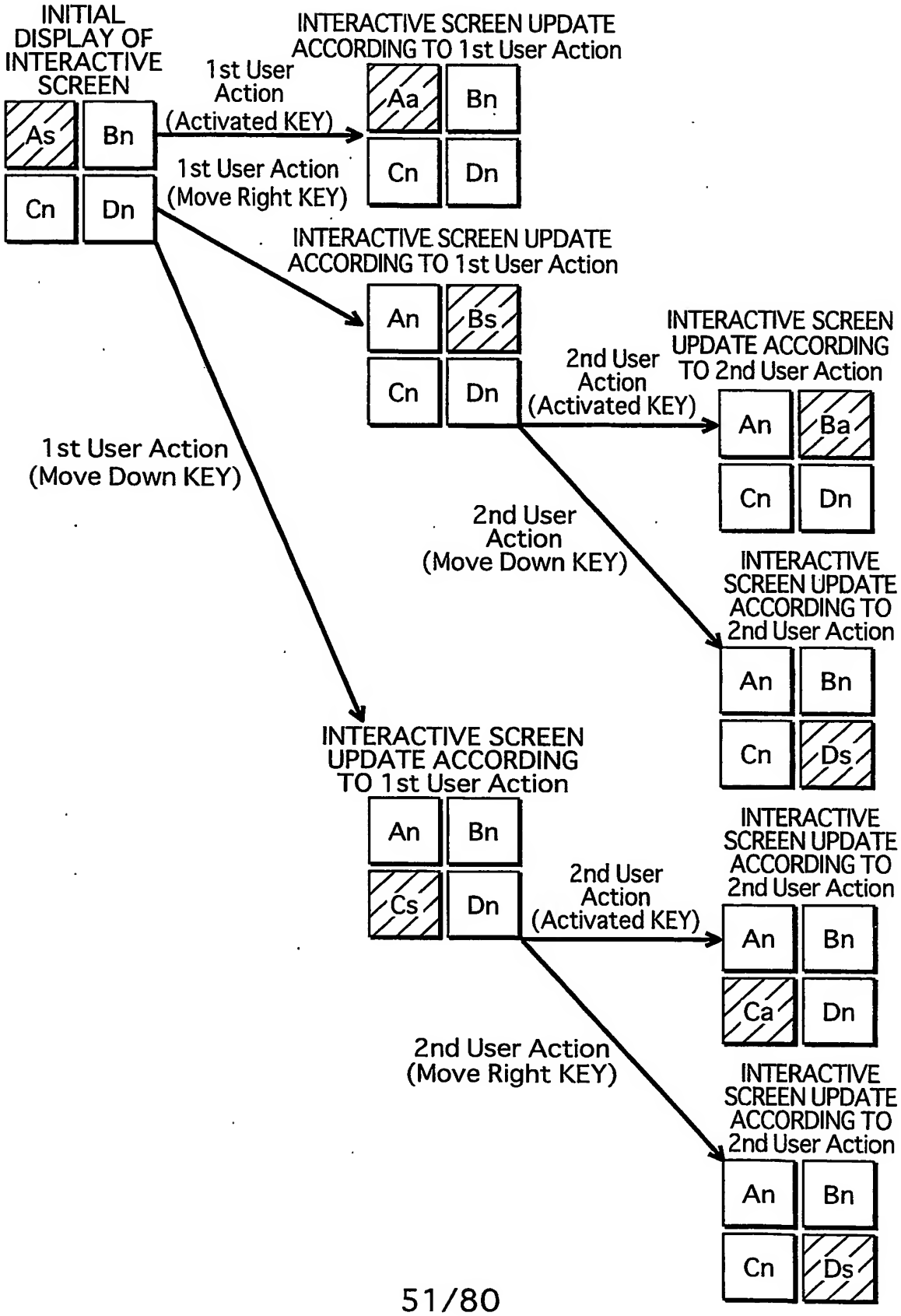

INITIAL DISPLAY OF INTERACTIVE SCREEN
As
Bn
Cn
Dn
1st User Action (Activated KEY)
INTERACTIVE SCREEN UPDATE ACCORDING TO 1st User Action
Aa
Bn
Cn
Dn
1st User Action (Move Right KEY)
INTERACTIVE SCREEN UPDATE ACCORDING TO 1st User Action
An
Bs
Cn
Dn
2nd User Action (Activated KEY)
INTERACTIVE SCREEN UPDATE ACCORDING TO 2nd User Action
An
Ba
Cn
Dn
1st User Action (Move Down KEY)
2nd User Action (Move Down KEY)
INTERACTIVE SCREEN UPDATE ACCORDING TO 2nd User Action
An
Bn
Cn
Ds
INTERACTIVE SCREEN UPDATE ACCORDING TO 1st User Action
An
Bn
Cs
Dn
2nd User Action (Activated KEY)
INTERACTIVE SCREEN UPDATE ACCORDING TO 2nd User Action
An
Bn
Ca
Dn
2nd User Action (Move Right KEY)
INTERACTIVE SCREEN UPDATE ACCORDING TO 2nd User Action
An
Bn
Cn
Ds

FIG. 52

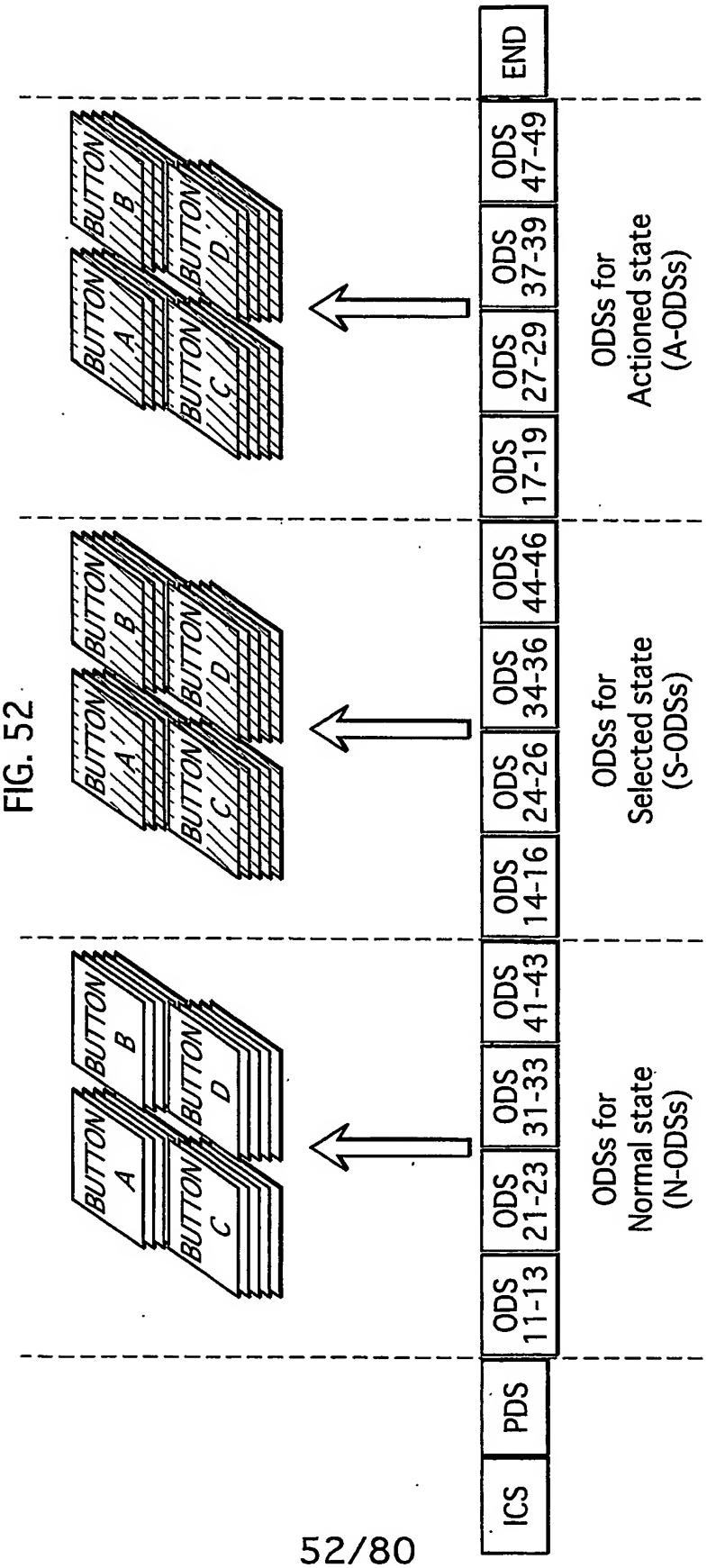

ICS
PDS
ODS 11-13
ODS 21-23
ODS 31-33
ODS 41-43
ODS 14-16
ODS 24-26
ODS 34-36
ODS 44-46
ODS 17-19
ODS 27-29
ODS 37-39
ODS 47-49
END
BUTTON A
BUTTON B
BUTTON C
BUTTON D
ODSs for Normal state (N-ODSs)
ODSs for Selected state (S-ODSs)
ODSs for Actioned state (A-ODSs)

FIG. 53

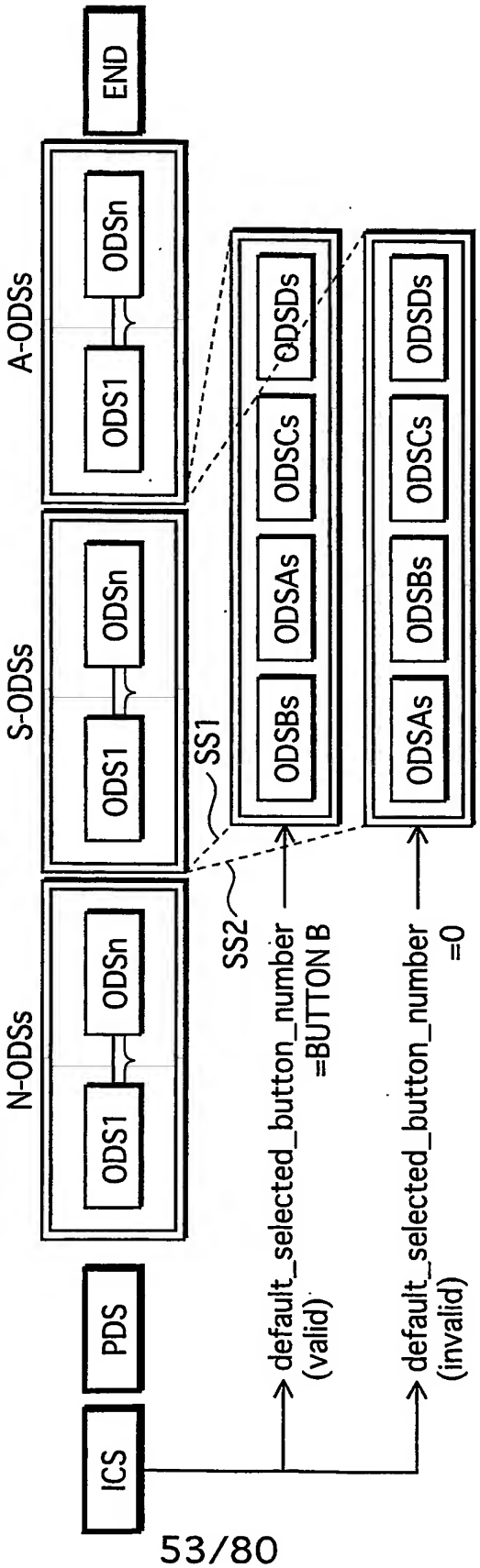

ICS
PDS
N-ODSs
ODS1
ODSn
S-ODSs
ODS1
ODSn
A-ODSs
ODS1
ODSn
END
SS1
SS2
ODSBs
ODSAs
ODSCs
ODSDs
default_selected_button_number
=BUTTON B
(valid)
default_selected_button_number
=0
(invalid)

FIG. 54A

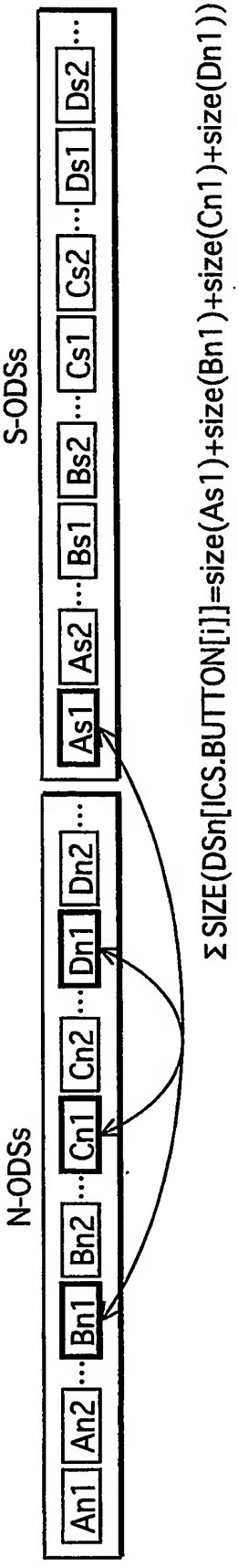


FIG. 54B

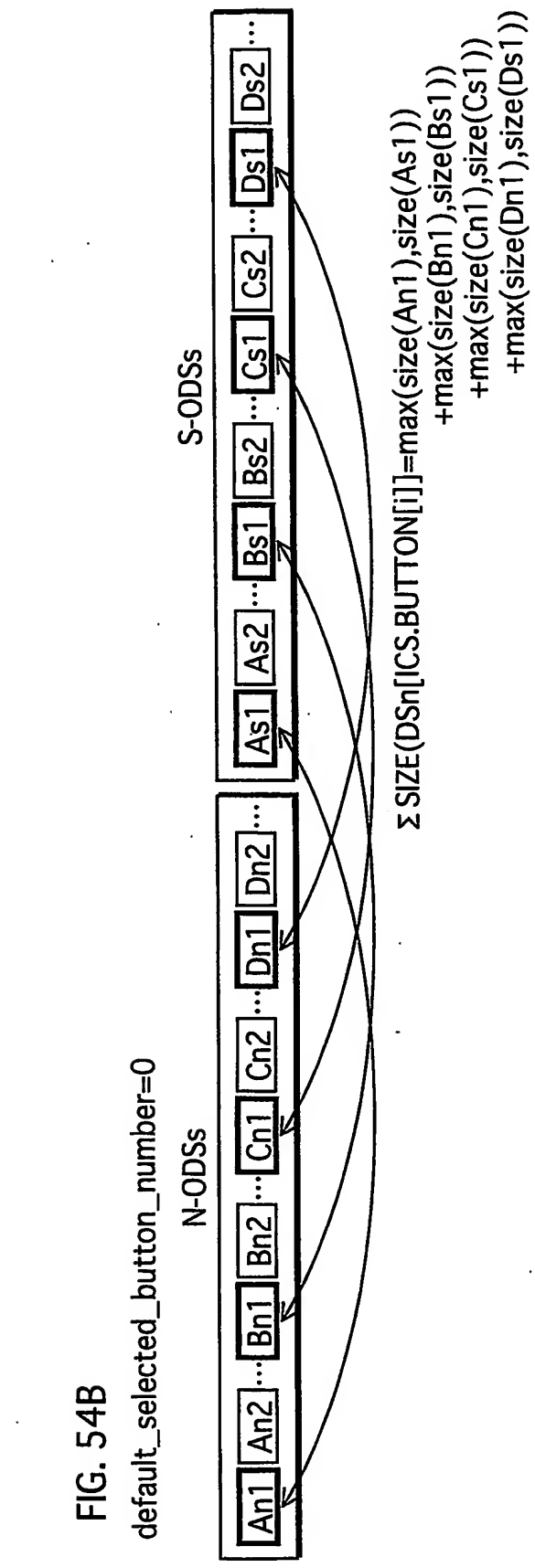


FIG. 55

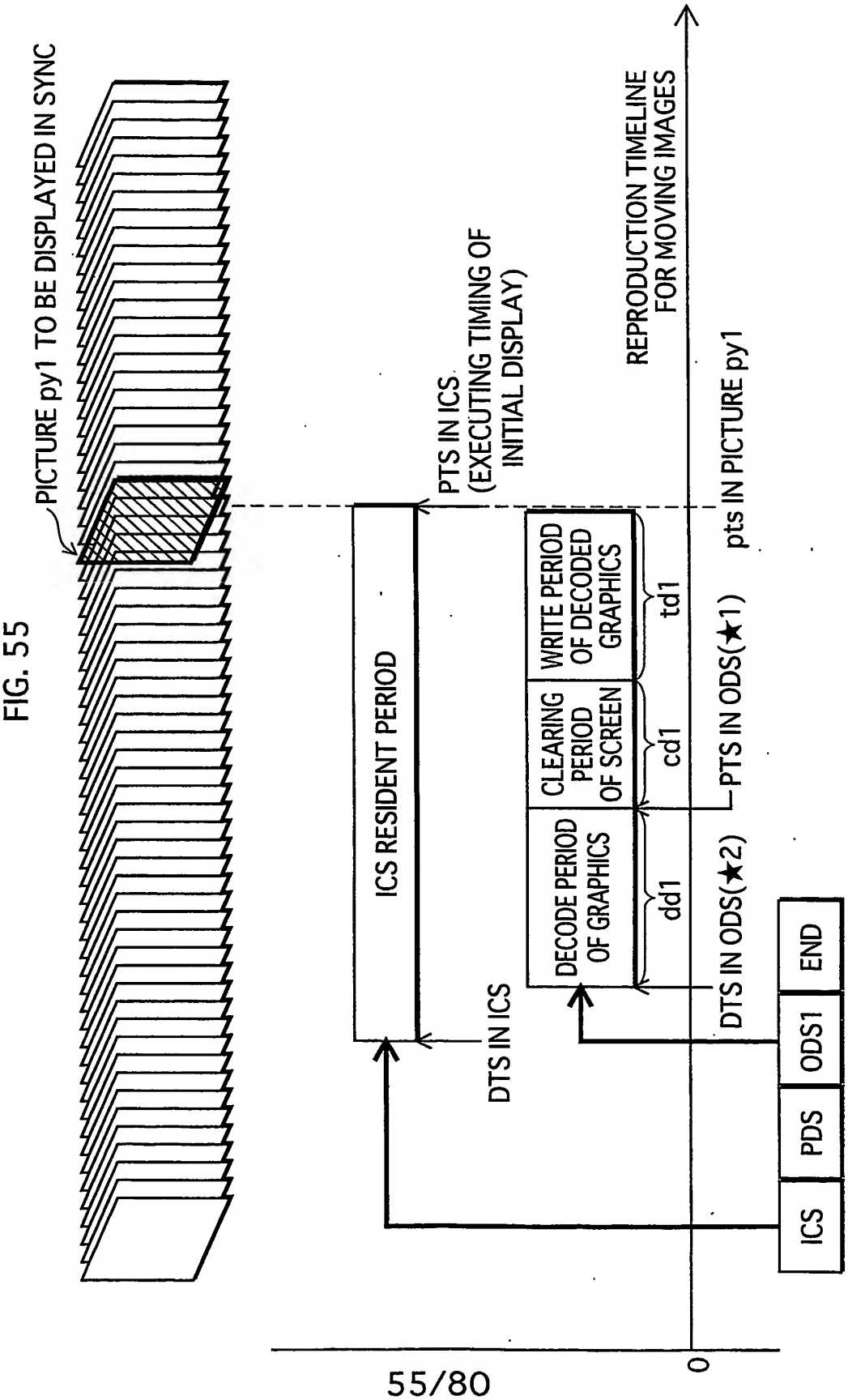

PICTURE py1 TO BE DISPLAYED IN SYNC
PTS IN ICS
(EXECUTING TIMING OF INITIAL DISPLAY)
REPRODUCTION TIMELINE FOR MOVING IMAGES
pts IN PICTURE py1
ICS RESIDENT PERIOD
WRITE PERIOD OF DECODED GRAPHICS
CLEARING PERIOD OF SCREEN
DECODE PERIOD OF GRAPHICS
td1
cd1
dd1
PTS IN ODS(★1)
DTS IN ODS(★2)
DTS IN ICS
ICS
PDS
ODS1
END
0

FIG. 56

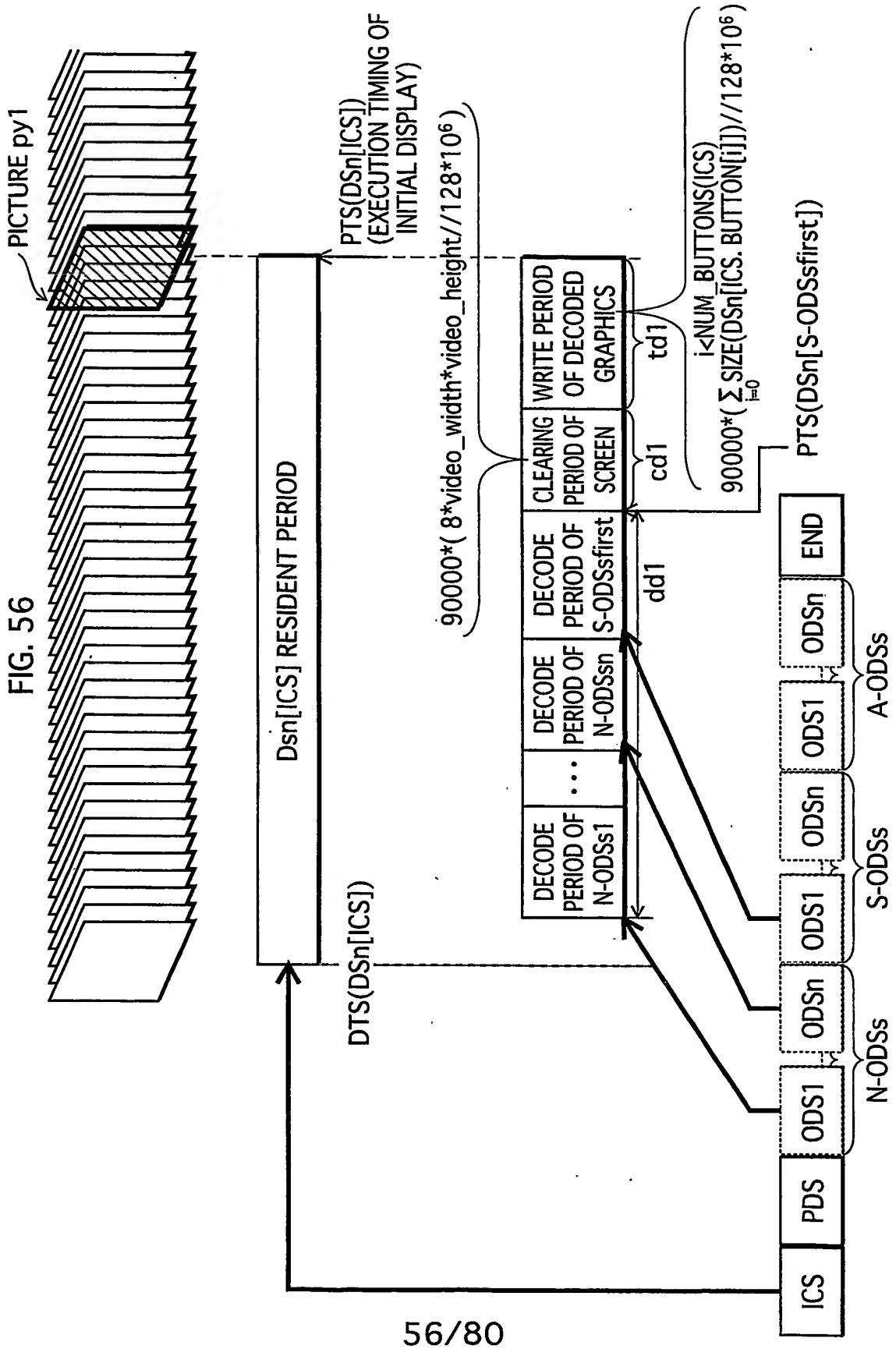

PICTURE py1
PTS(DSn[ICS])
(EXECUTION TIMING OF
INITIAL DISPLAY)
Dsn[ICS] RESIDENT PERIOD
DTS(DSn[ICS])
90000*(8*video_width*video_height//128*10^6)
DECODE PERIOD OF N-ODSs1
...
DECODE PERIOD OF N-ODSsn
DECODE PERIOD OF S-ODSsfirst
CLEARING PERIOD OF SCREEN
WRITE PERIOD OF DECODED GRAPHICS
dd1
cd1
td1
i<NUM_BUTTONS(ICS)
90000*(Σ SIZE(DSn[ICS. BUTTON[i]])//128*10^6)
i=0
PTS(DSn[S-ODSsfirst])
ICS
PDS
ODS1
ODSn
ODS1
ODSn
ODS1
ODSn
END
N-ODSs
S-ODSs
A-ODSs

FIG. 57

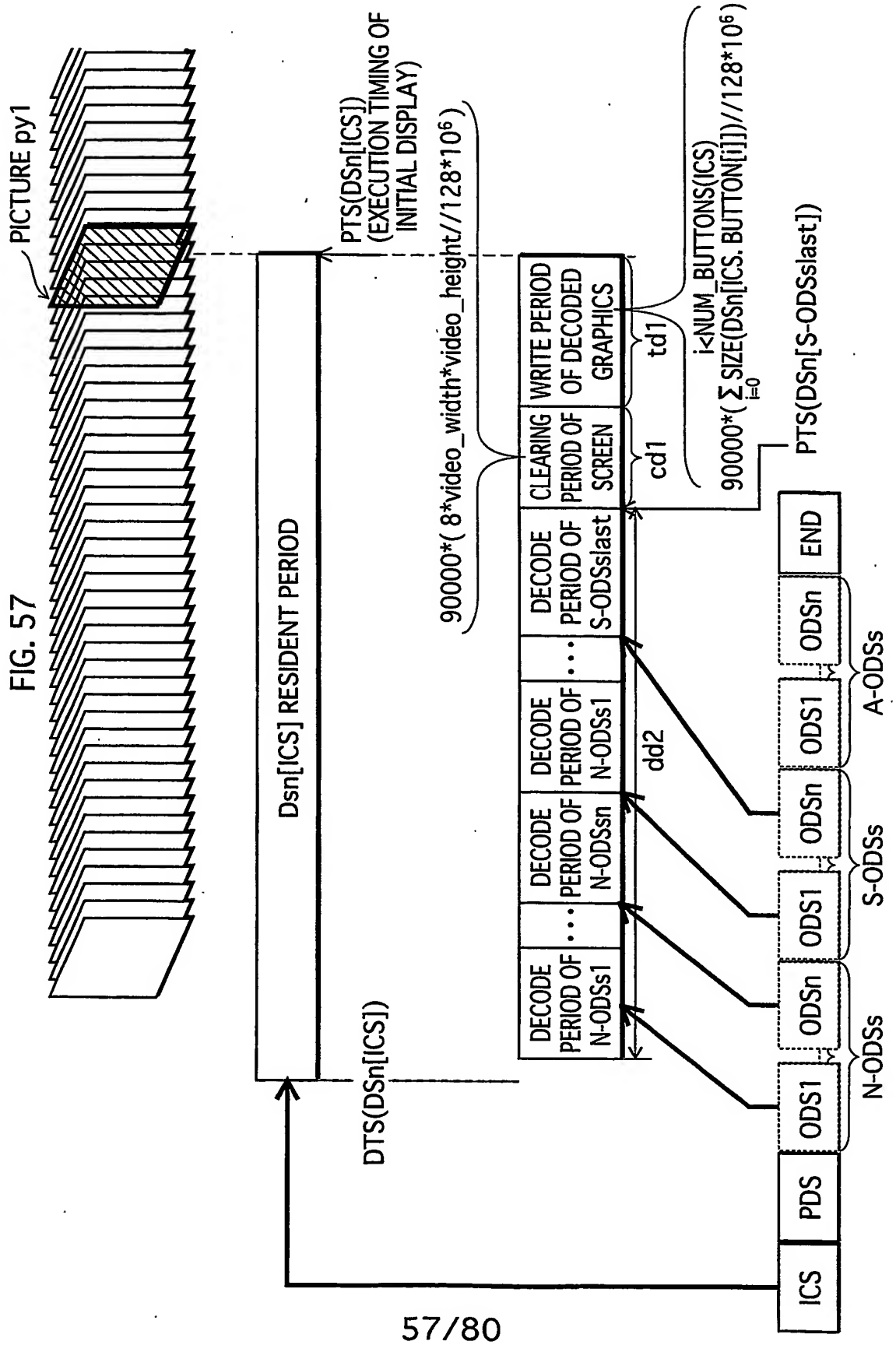

PICTURE py1
PTS(DSn[ICS])
(EXECUTION TIMING OF INITIAL DISPLAY)
Dsn[ICS] RESIDENT PERIOD
DTS(DSn[ICS])
90000*(8*video_width*video_height//128*10^6)
DECODE PERIOD OF N-ODSs1
DECODE PERIOD OF N-ODSsn
DECODE PERIOD OF S-ODSslast
CLEARING PERIOD OF SCREEN
WRITE PERIOD OF DECODED GRAPHICS
cd1
td1
dd2
i<NUM_BUTTONS(ICS)
90000*(Σ SIZE(DSn[ICS. BUTTON[i]])//128*10^6)
PTS(DSn[S-ODSslast])
ICS
PDS
ODS1
ODSn
ODS1
ODSn
ODS1
ODSn
END
N-ODSs
S-ODSs
A-ODSs

FIG. 58

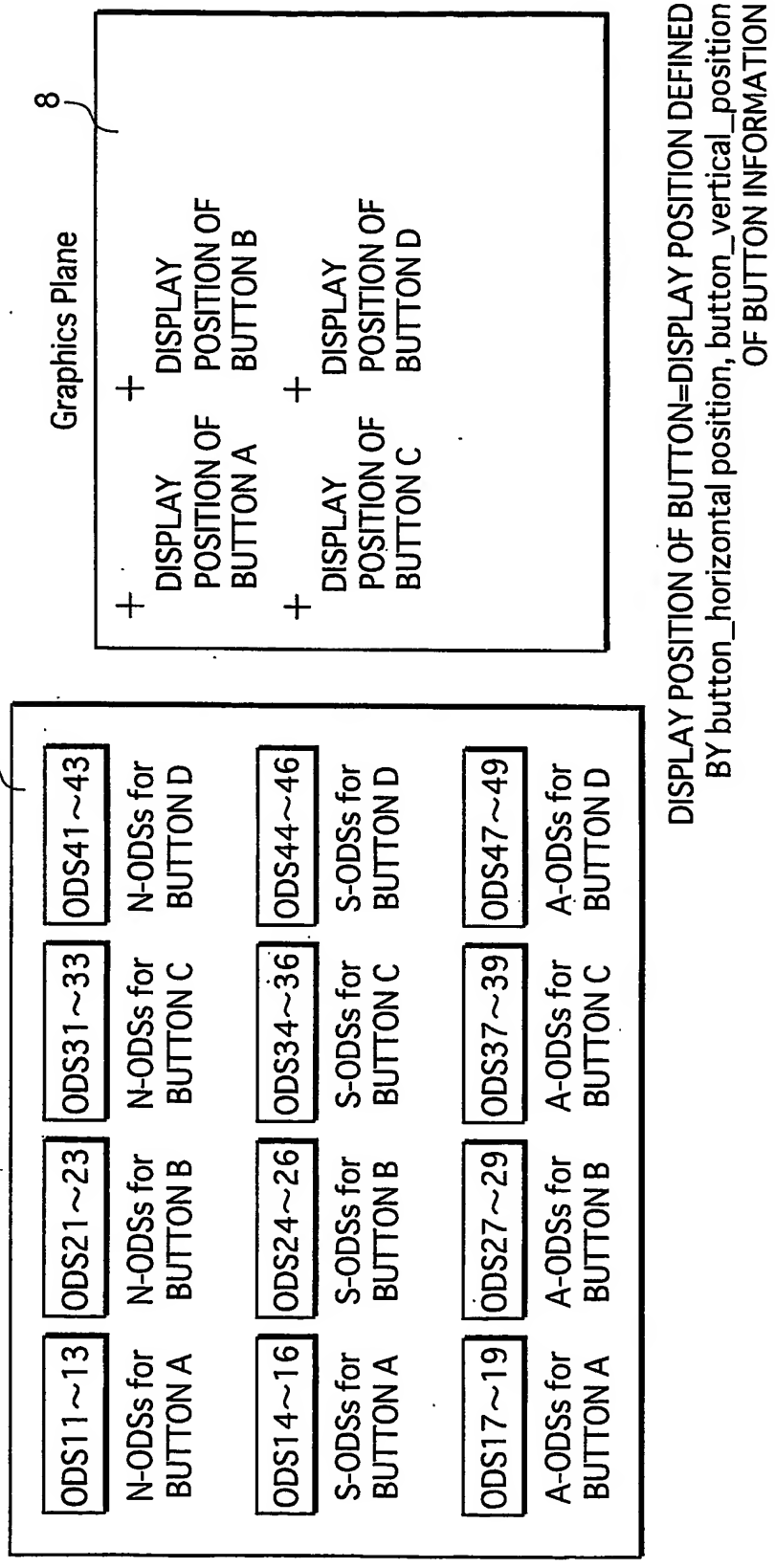

Object Buffer
15
ODS11~13
N-ODSs for BUTTON A
ODS21~23
N-ODSs for BUTTON B
ODS31~33
N-ODSs for BUTTON C
ODS41~43
N-ODSs for BUTTON D
ODS14~16
S-ODSs for BUTTON A
ODS24~26
S-ODSs for BUTTON B
ODS34~36
S-ODSs for BUTTON C
ODS44~46
S-ODSs for BUTTON D
ODS17~19
A-ODSs for BUTTON A
ODS27~29
A-ODSs for BUTTON B
ODS37~39
A-ODSs for BUTTON C
ODS47~49
A-ODSs for BUTTON D
Graphics Plane
8
+ DISPLAY POSITION OF BUTTON A
+ DISPLAY POSITION OF BUTTON B
+ DISPLAY POSITION OF BUTTON C
+ DISPLAY POSITION OF BUTTON D
DISPLAY POSITION OF BUTTON=DISPLAY POSITION DEFINED BY button_horizontal position, button_vertical_position OF BUTTON INFORMATION

FIG. 59
WRITE OPERATION OF Graphics Controller AT INITIAL DISPLAY

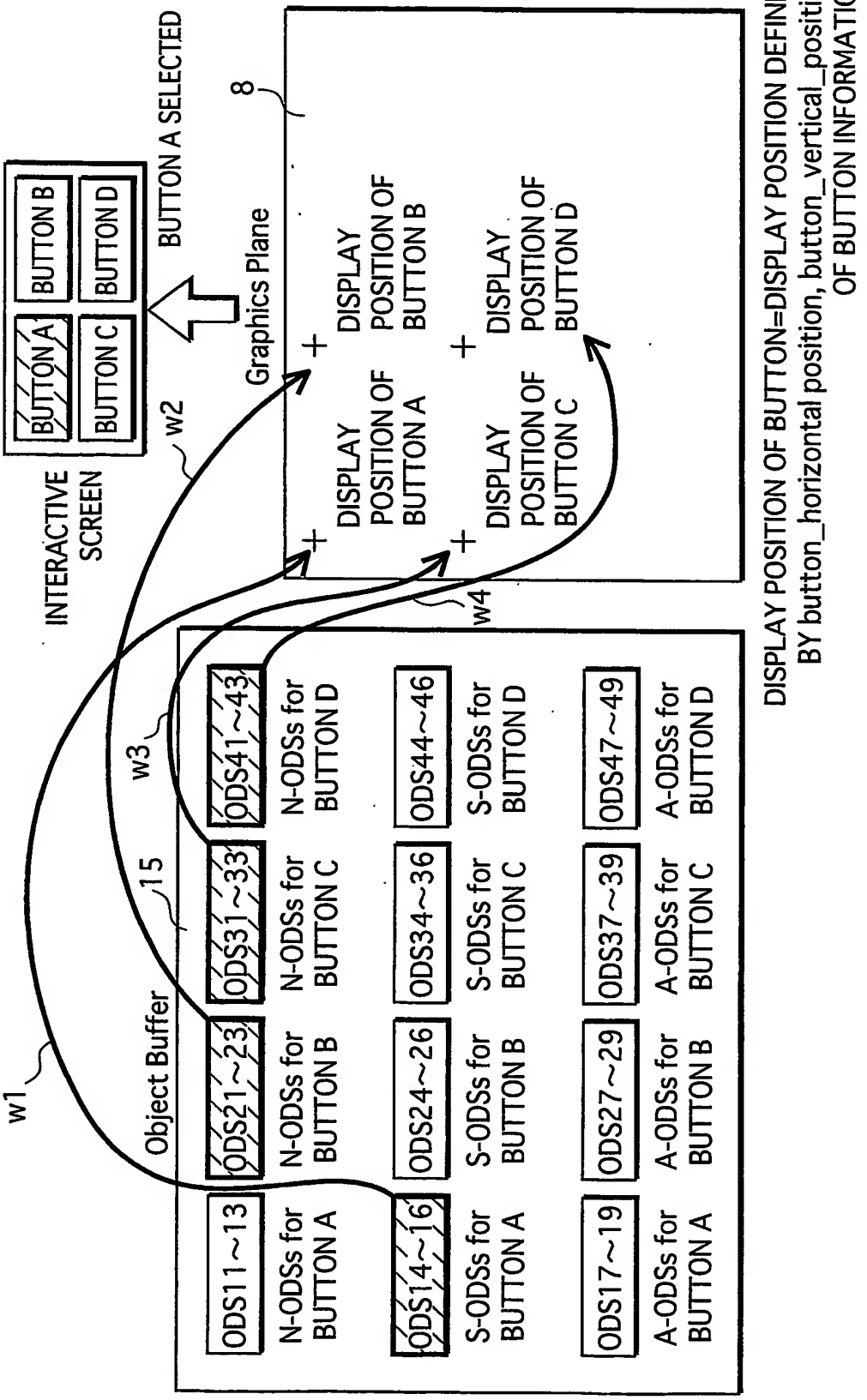
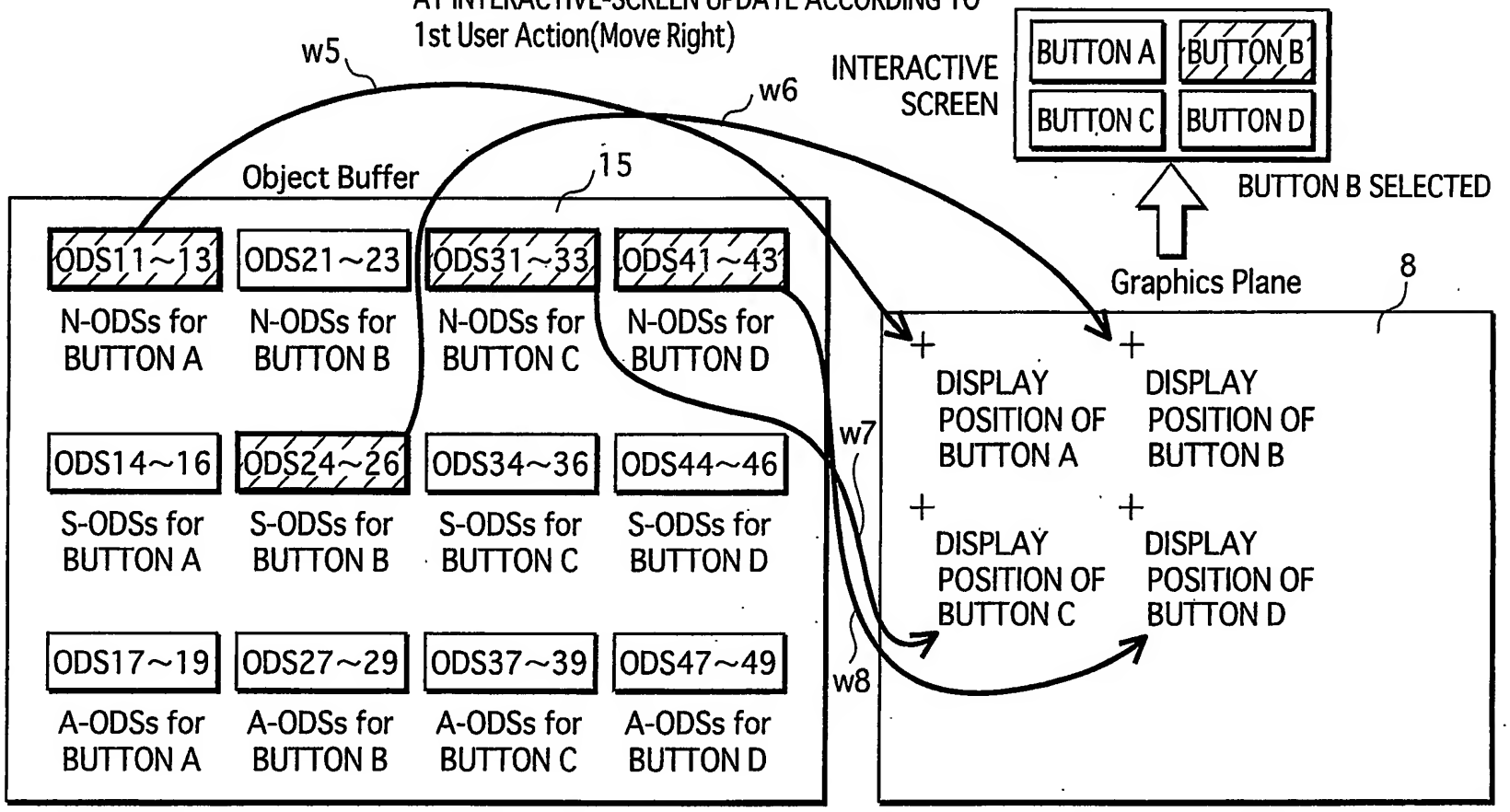


FIG. 60

WRITE OPERATION OF Graphics Controller
AT INTERACTIVE-SCREEN UPDATE ACCORDING TO
1st User Action(Move Right)



DISPLAY POSITION OF BUTTON=DISPLAY POSITION DEFINED
BY button_horizontal position, button_vertical_position
OF BUTTON INFORMATION

FIG. 61

WRITE OPERATION OF Graphics Controller
AT INTERACTIVE-SCREEN UPDATE ACCORDING TO
1st User Action(Move Down)

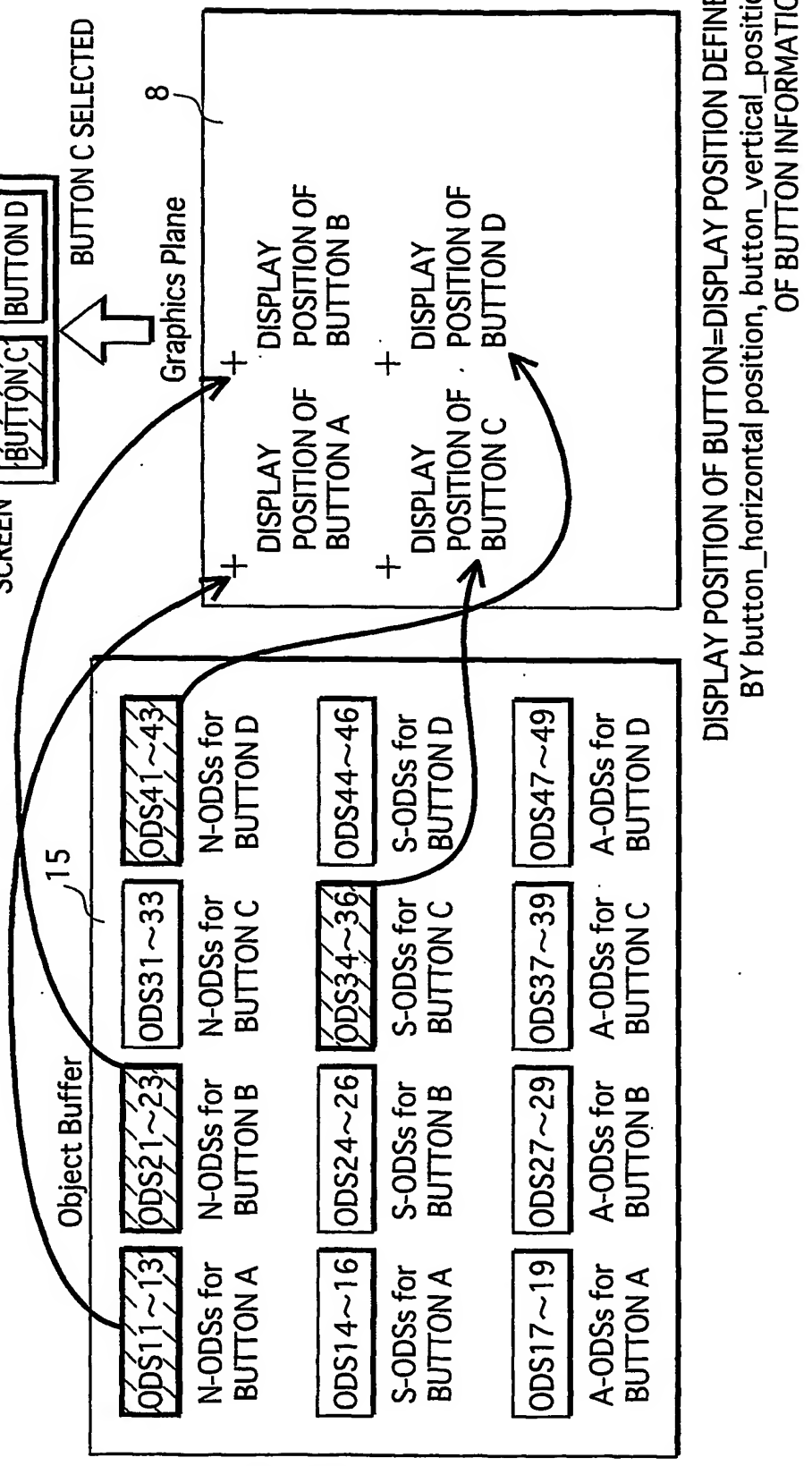


FIG. 62
WRITE OPERATION OF Graphics Controller
AT INTERACTIVE-SCREEN UPDATE ACCORDING TO
1st User Action(Activated)

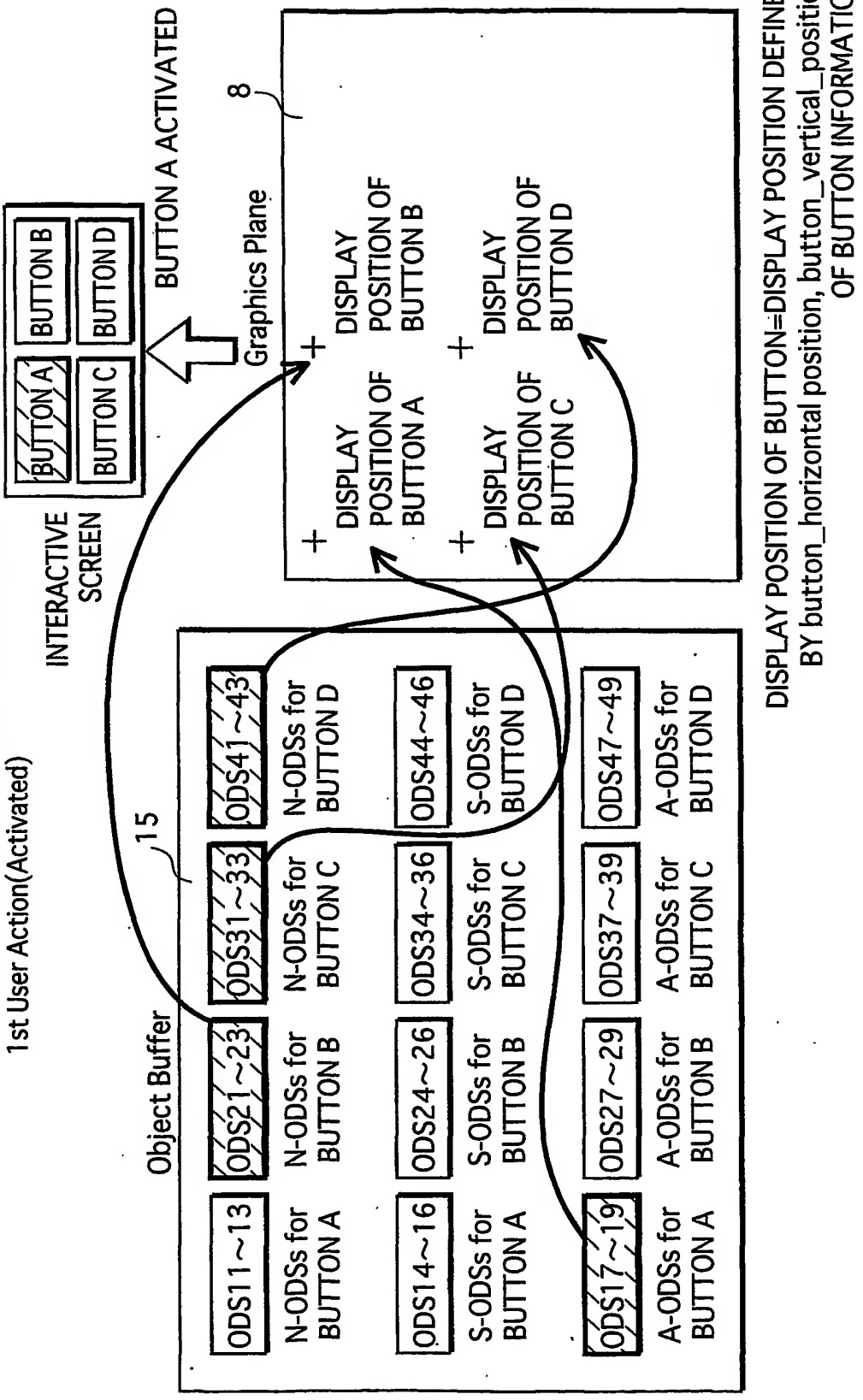
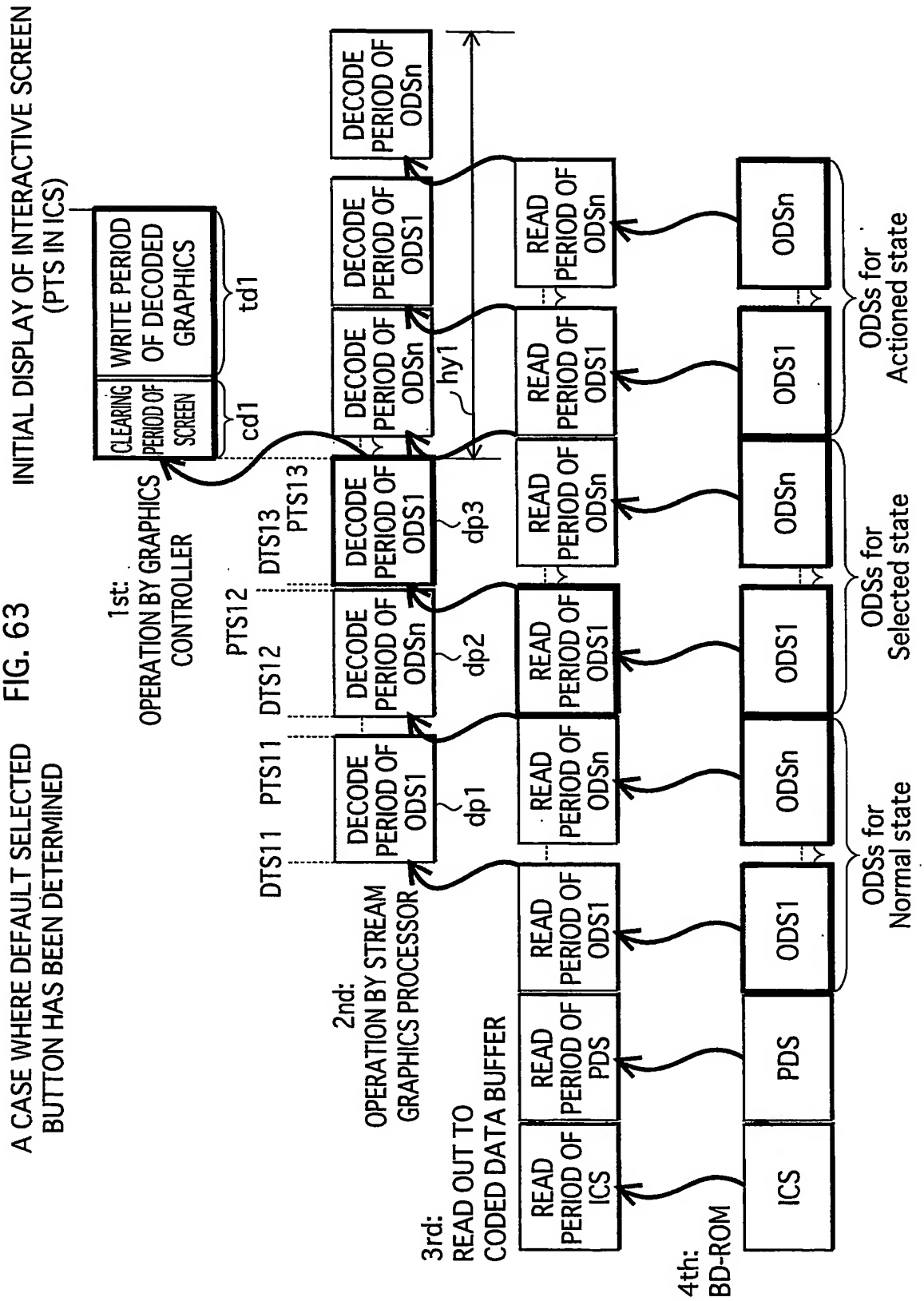


FIG. 63


A CASE WHERE DEFAULT SELECTED BUTTON HAS BEEN DETERMINED
INITIAL DISPLAY OF INTERACTIVE SCREEN (PTS IN ICS)
1st: OPERATION BY GRAPHICS CONTROLLER
2nd: OPERATION BY STREAM GRAPHICS PROCESSOR
3rd: READ OUT TO CODED DATA BUFFER
4th: BD-ROM
CLEARING PERIOD OF SCREEN
WRITE PERIOD OF DECODED GRAPHICS
cd1
td1
DTS11
PTS11
DTS12
PTS12
DTS13
PTS13
dp1
dp2
dp3
hy1
READ PERIOD OF ICS
READ PERIOD OF PDS
READ PERIOD OF ODS1
DECODE PERIOD OF ODS1
DECODE PERIOD OF ODSn
ICS
PDS
ODS1
ODSn
ODSs for Normal state
ODSs for Selected state
ODSs for Actioned state

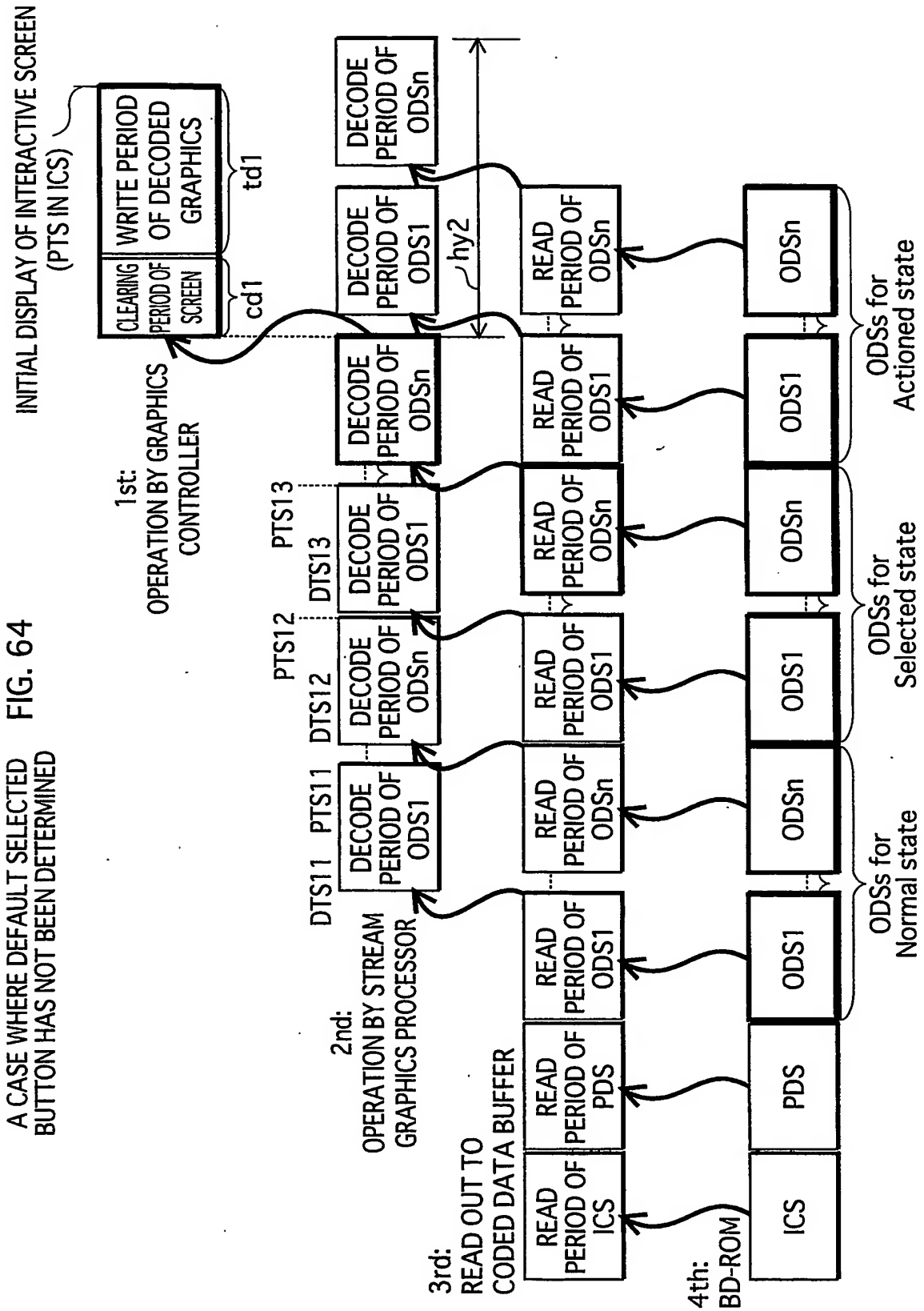

A CASE WHERE DEFAULT SELECTED BUTTON HAS NOT BEEN DETERMINED
INITIAL DISPLAY OF INTERACTIVE SCREEN (PTS IN ICS)
CLEARING PERIOD OF SCREEN
WRITE PERIOD OF DECODED GRAPHICS
cd1
td1
1st: OPERATION BY GRAPHICS CONTROLLER
PTS13
DTS13
PTS12
DTS12
PTS11
DTS11
DECODE PERIOD OF ODS1
DECODE PERIOD OF ODSn
hy2
2nd: OPERATION BY STREAM GRAPHICS PROCESSOR
READ PERIOD OF ODS1
READ PERIOD OF ODSn
3rd: READ OUT TO CODED DATA BUFFER
READ PERIOD OF ICS
READ PERIOD OF PDS
4th: BD-ROM
ICS
PDS
ODS1
ODSn
ODSs for Normal state
ODSs for Selected state
ODSs for Actioned state

FIG. 64

FIG. 65

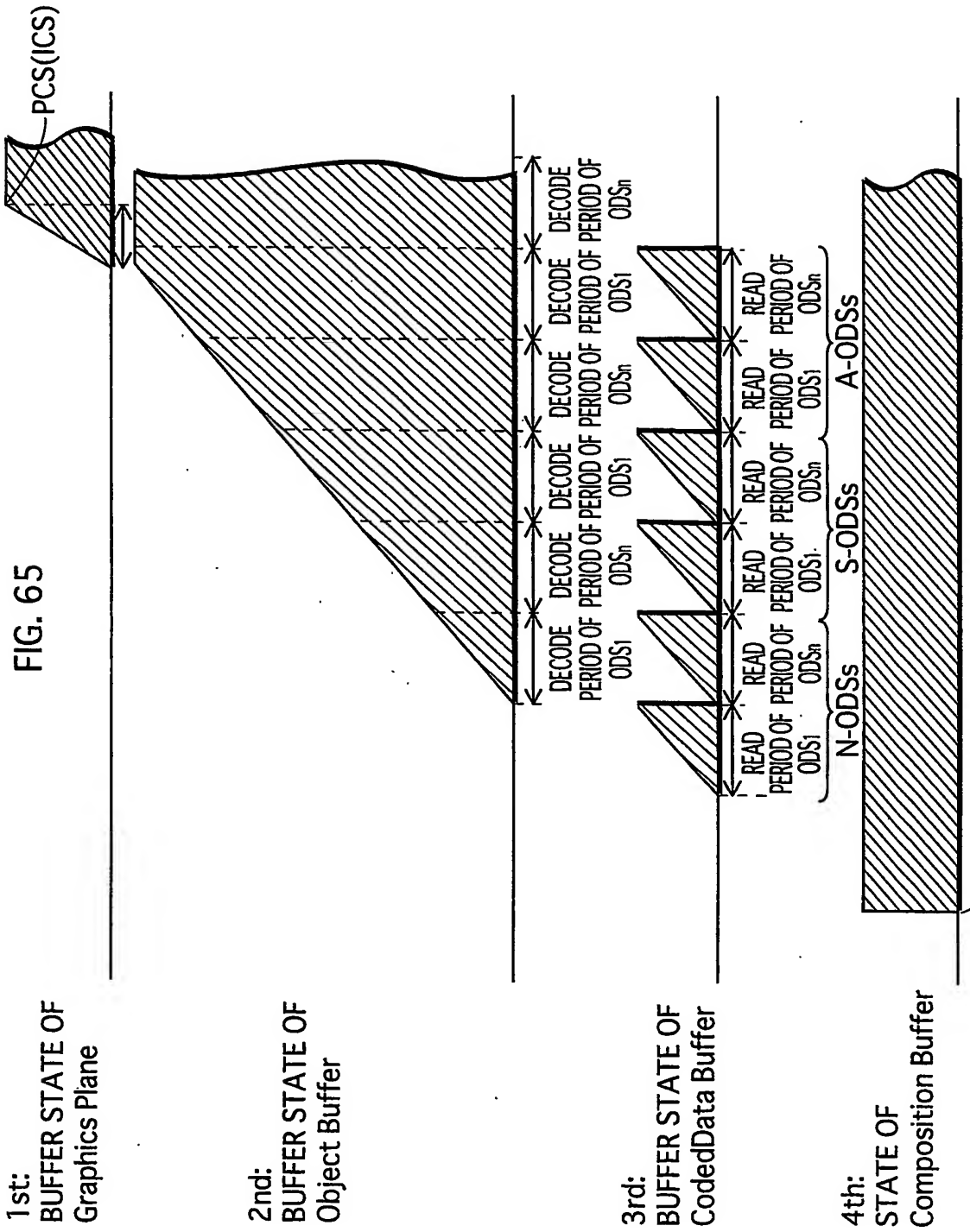

1st:
BUFFER STATE OF
Graphics Plane
PCS(ICS)
2nd:
BUFFER STATE OF
Object Buffer
DECODE PERIOD OF ODS1
DECODE PERIOD OF ODSn
3rd:
BUFFER STATE OF
CodedData Buffer
READ PERIOD OF ODS1
READ PERIOD OF ODSn
N-ODSs
S-ODSs
A-ODSs
4th:
STATE OF
Composition Buffer
DTS(ICS)

FIG. 66

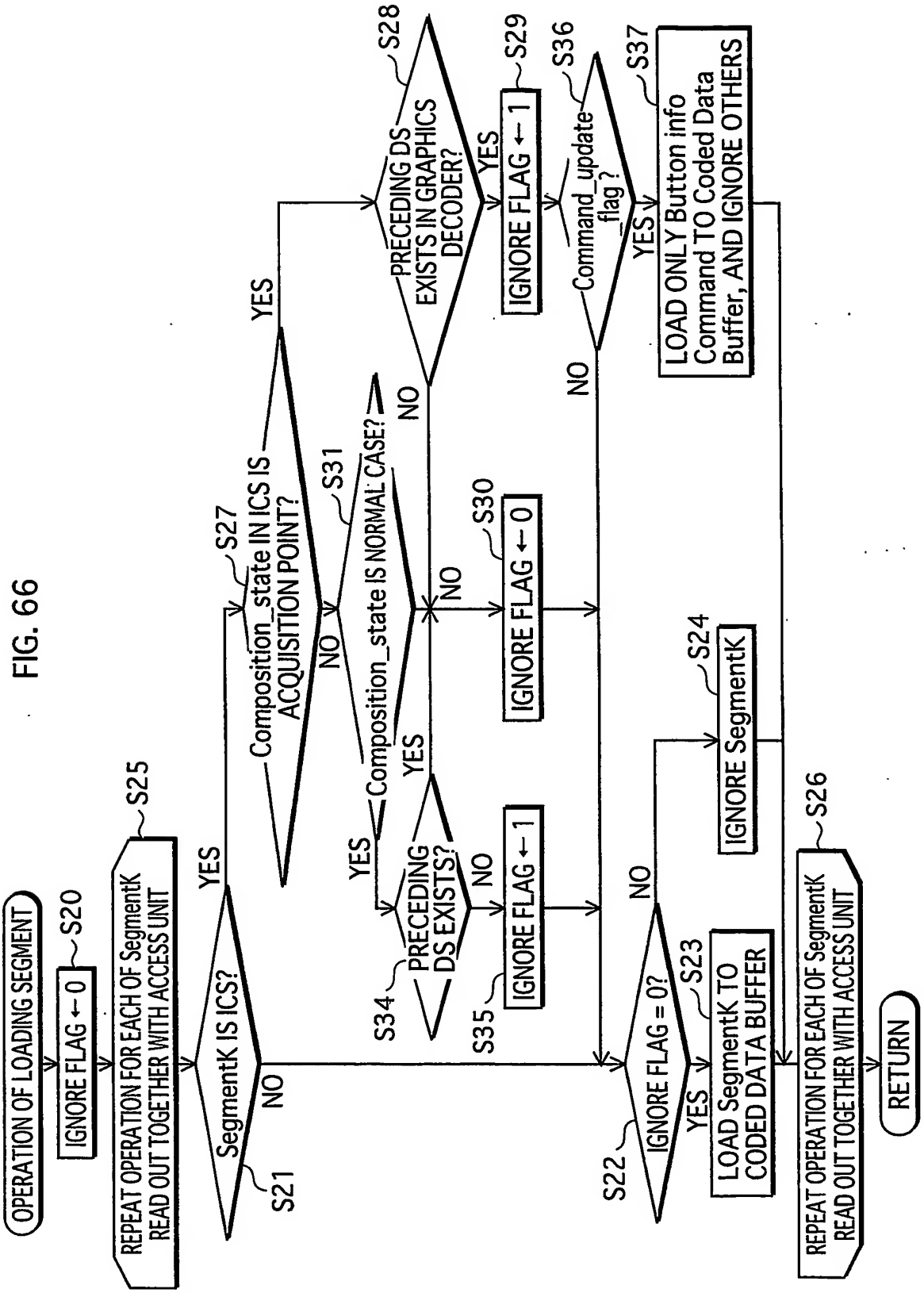

OPERATION OF LOADING SEGMENT
IGNORE FLAG ← 0
S20
REPEAT OPERATION FOR EACH OF SegmentK READ OUT TOGETHER WITH ACCESS UNIT
S25
SegmentK IS ICS?
YES
NO
S21
Composition_state IN ICS IS ACQUISITION POINT?
S27
YES
NO
Composition_state IS NORMAL CASE?
S31
YES
NO
PRECEDING DS EXISTS?
S34
YES
NO
IGNORE FLAG ← 1
S35
PRECEDING DS EXISTS IN GRAPHICS DECODER?
S28
YES
NO
IGNORE FLAG ← 1
S29
IGNORE FLAG ← 0
S30
Command_update flag ?
S36
YES
NO
LOAD ONLY Button info Command TO Coded Data Buffer, AND IGNORE OTHERS
S37
IGNORE FLAG = 0?
S22
YES
NO
LOAD SegmentK TO CODED DATA BUFFER
S23
IGNORE SegmentK
S24
REPEAT OPERATION FOR EACH OF SegmentK READ OUT TOGETHER WITH ACCESS UNIT
S26
RETURN

FIG. 67

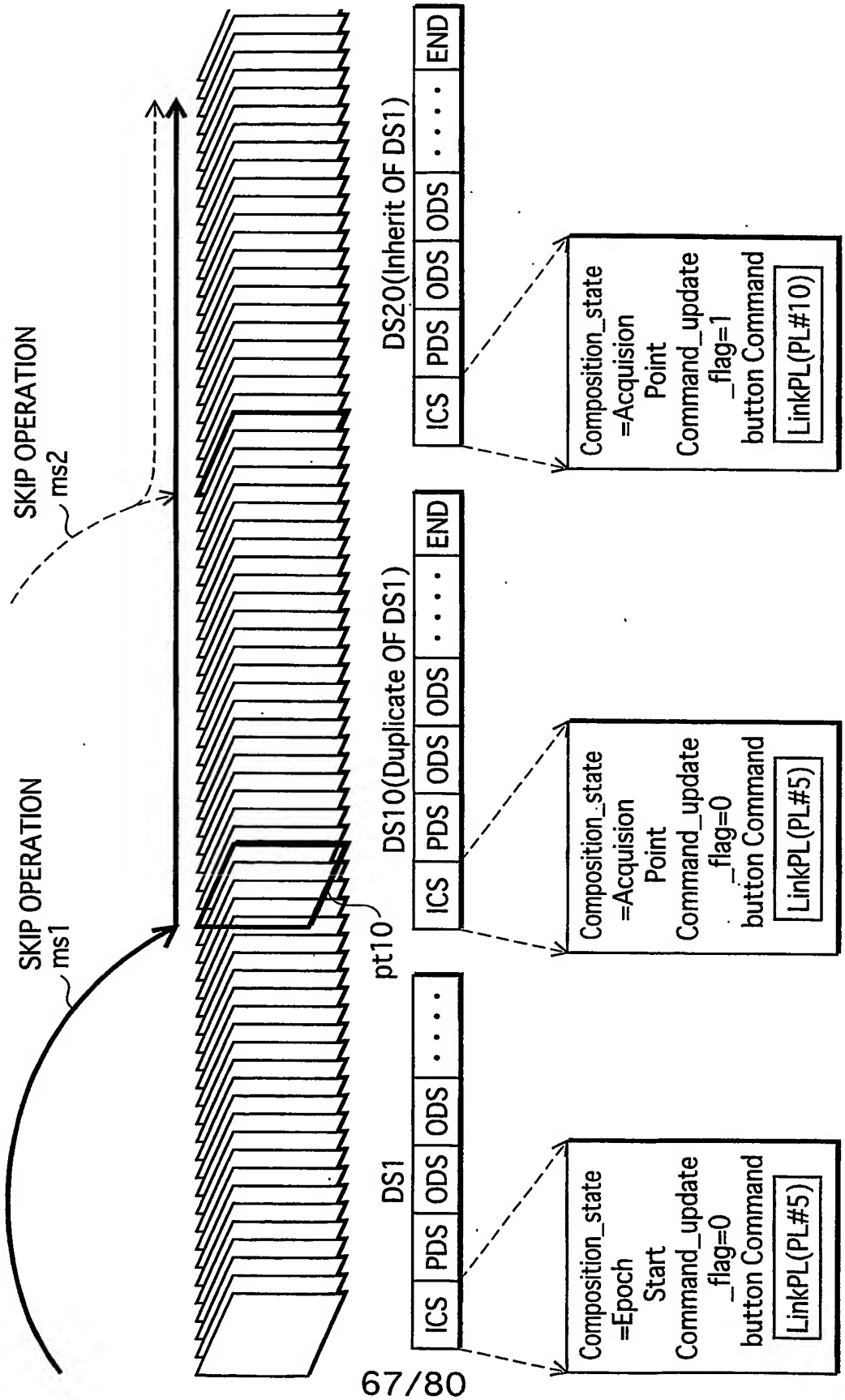

SKIP OPERATION
ms1
SKIP OPERATION
ms2
pt10
DS1
ICS
PDS
ODS
ODS
• • • •
DS10(Duplicate OF DS1)
ICS
PDS
ODS
ODS
• • • •
END
DS20(Inherit OF DS1)
ICS
PDS
ODS
ODS
• • • •
END
Composition_state
=Epoch
Start
Command_update
_flag=0
button Command
LinkPL(PL#5)
Composition_state
=Acquision
Point
Command_update
_flag=0
button Command
LinkPL(PL#5)
Composition_state
=Acquision
Point
Command_update
_flag=1
button Command
LinkPL(PL#10)

FIG. 68

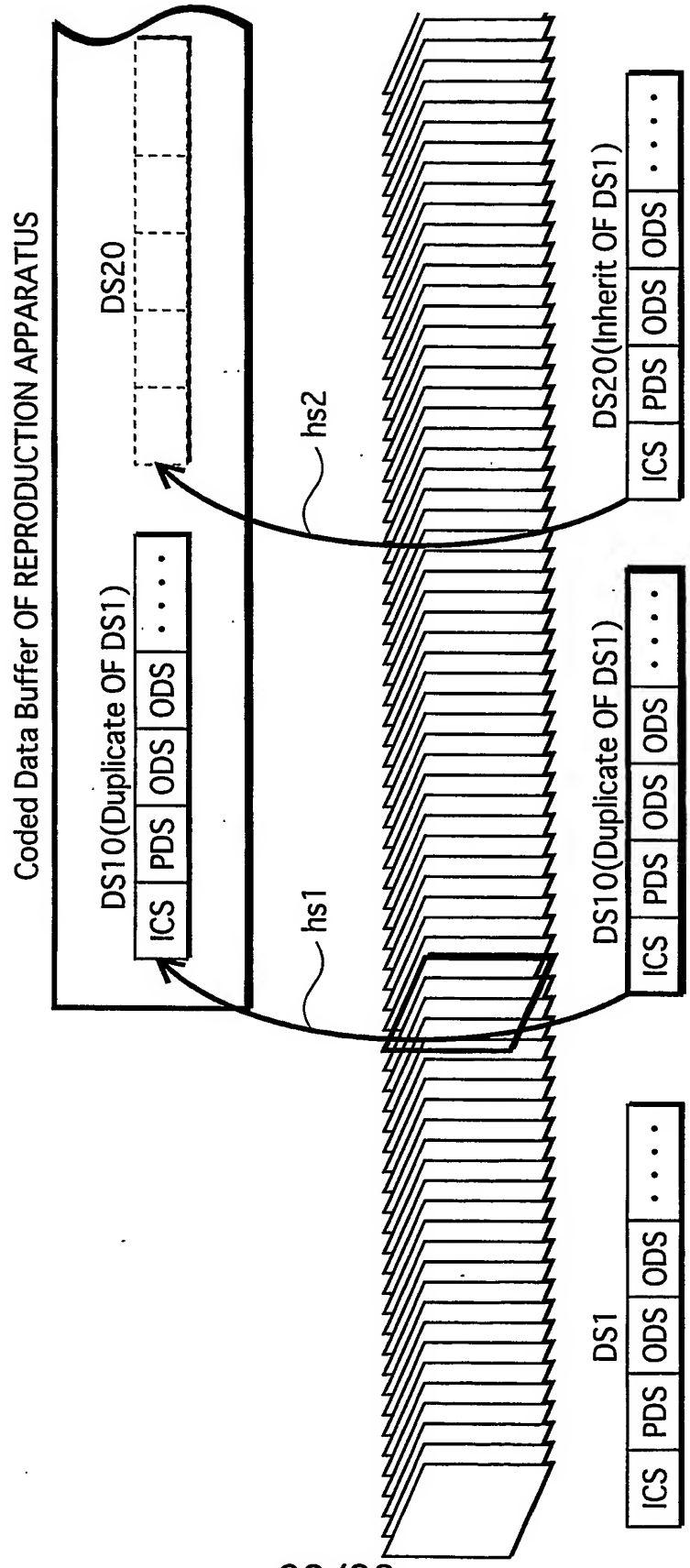

Coded Data Buffer OF REPRODUCTION APPARATUS
DS20
DS10(Duplicate OF DS1)
ICS
PDS
ODS
ODS
. . . .
hs1
hs2
DS1
DS10(Duplicate OF DS1)
DS20(Inherit OF DS1)

FIG. 69

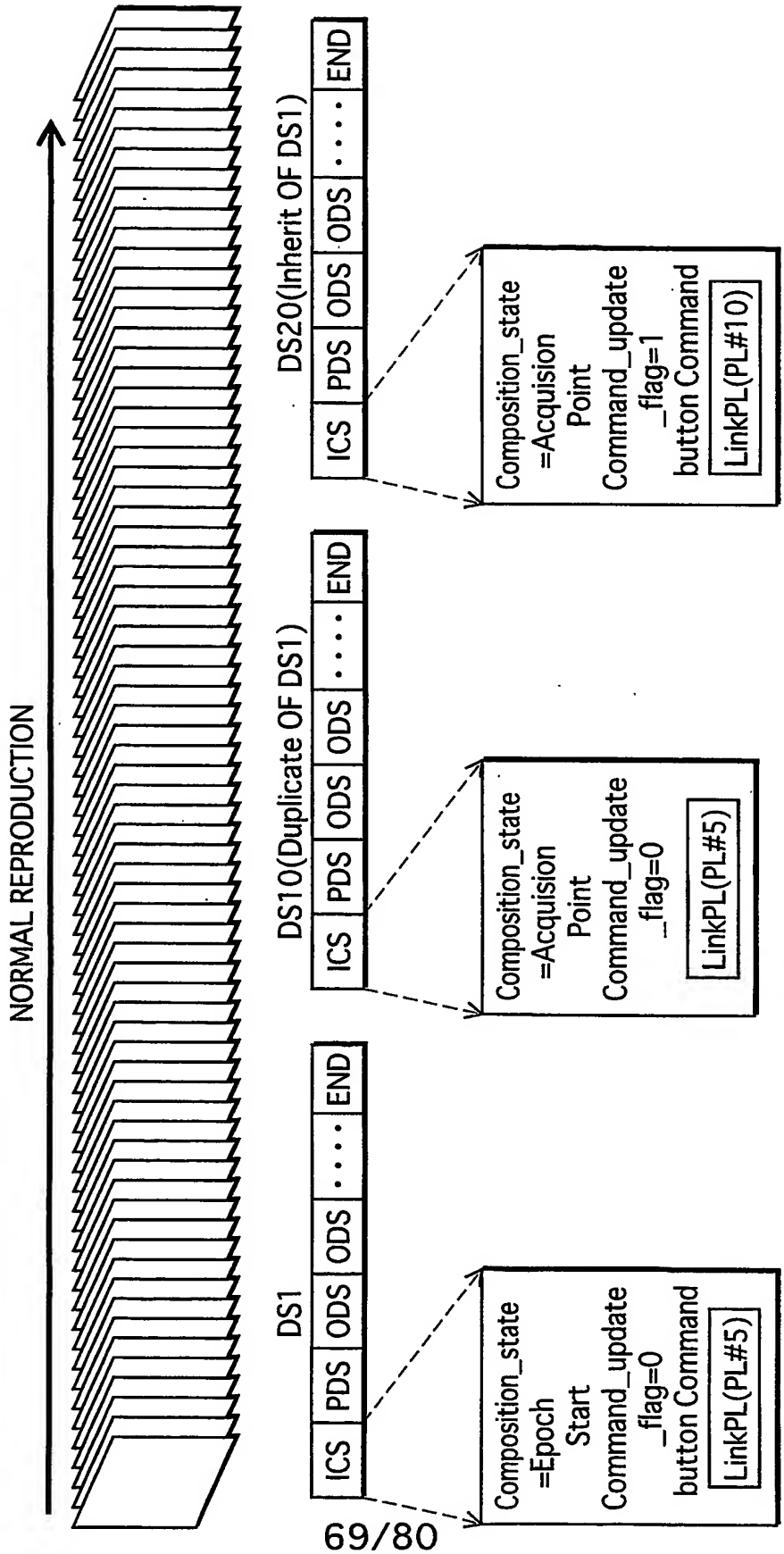

NORMAL REPRODUCTION
DS1
ICS
PDS
ODS
ODS
• • • •
END
Composition_state
=Epoch
Start
Command_update
_flag=0
button Command
LinkPL(PL#5)
DS10(Duplicate OF DS1)
ICS
PDS
ODS
ODS
• • • •
END
Composition_state
=Acquision
Point
Command_update
_flag=0
LinkPL(PL#5)
DS20(Inherit OF DS1)
ICS
PDS
ODS
ODS
• • • •
END
Composition_state
=Acquision
Point
Command_update
_flag=1
button Command
LinkPL(PL#10)

FIG. 70

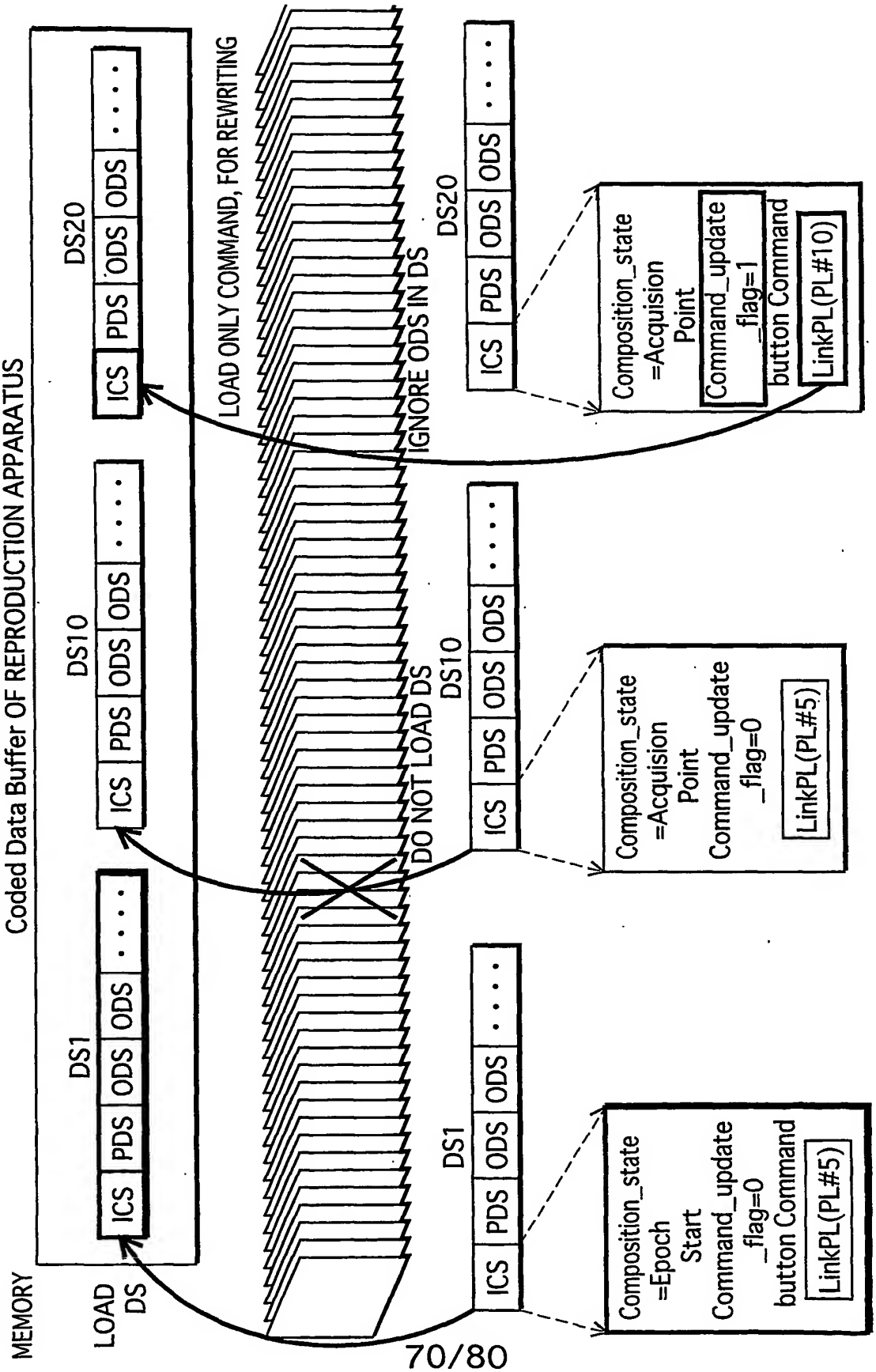

MEMORY
Coded Data Buffer OF REPRODUCTION APPARATUS
LOAD DS
DS1
ICS
PDS
ODS
ODS
DS10
ICS
PDS
ODS
ODS
DS20
ICS
PDS
ODS
ODS
LOAD ONLY COMMAND, FOR REWRITING
IGNORE ODS IN DS
DO NOT LOAD DS
Composition_state =Epoch Start
Command_update _flag=0
button Command
LinkPL(PL#5)
Composition_state =Acquision Point
Command_update _flag=0
LinkPL(PL#5)
Composition_state =Acquision Point
Command_update _flag=1
button Command
LinkPL(PL#10)

FIG.71

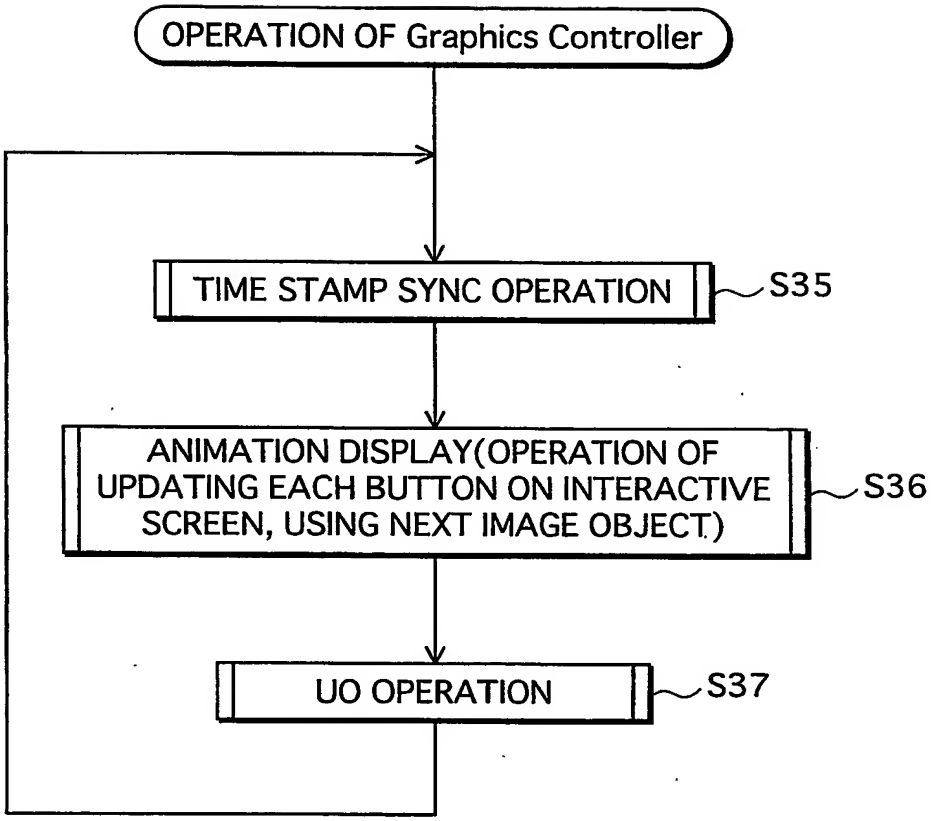
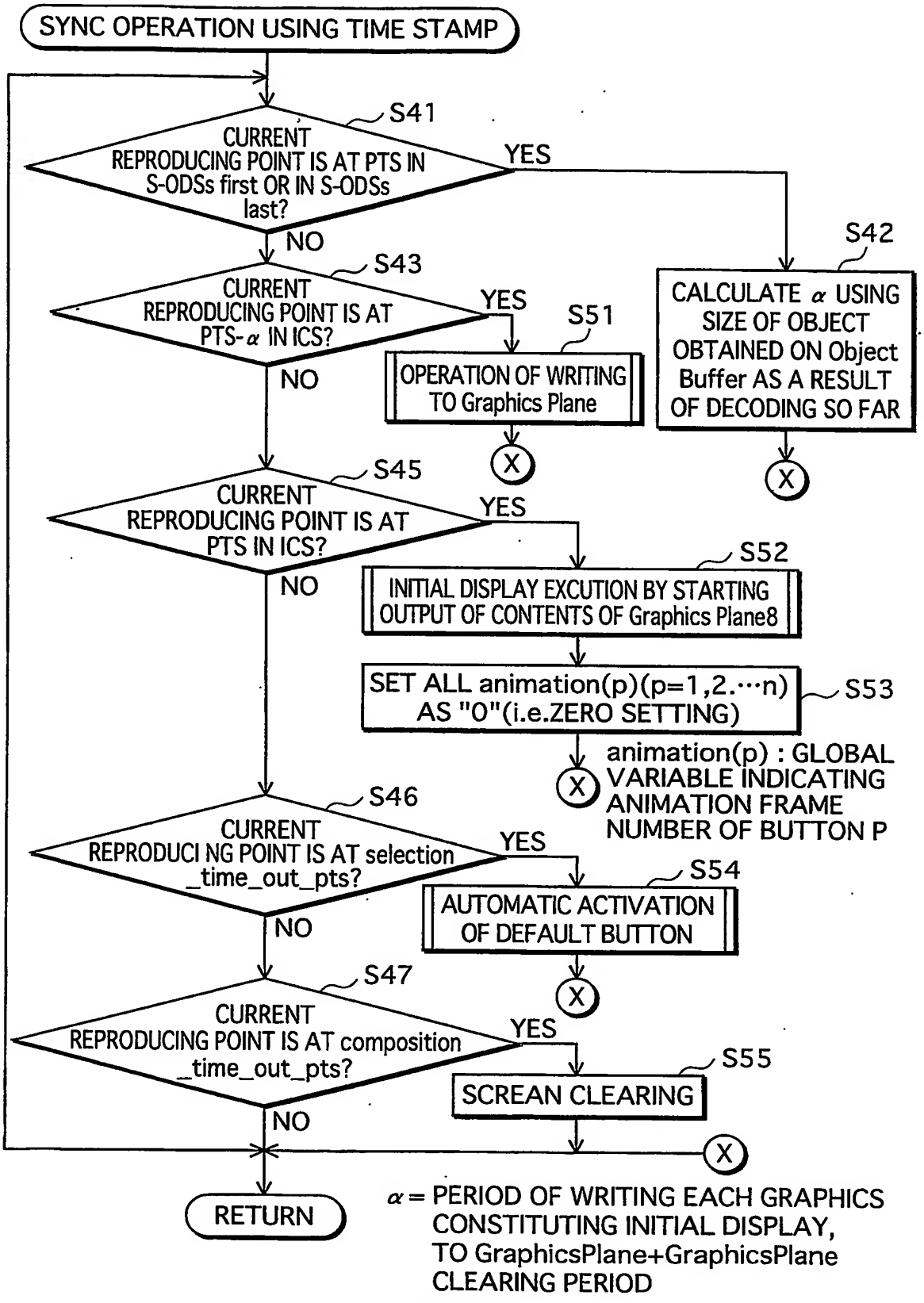

OPERATION OF Graphics Controller
TIME STAMP SYNC OPERATION
S35
ANIMATION DISPLAY(OPERATION OF UPDATING EACH BUTTON ON INTERACTIVE SCREEN, USING NEXT IMAGE OBJECT)
S36
UO OPERATION
S37

FIG.72


SYNC OPERATION USING TIME STAMP
S41
CURRENT REPRODUCING POINT IS AT PTS IN S-ODSs first OR IN S-ODSs last?
YES
NO
S42
CALCULATE α USING SIZE OF OBJECT OBTAINED ON Object Buffer AS A RESULT OF DECODING SO FAR
S43
CURRENT REPRODUCING POINT IS AT PTS-α IN ICS?
YES
NO
S51
OPERATION OF WRITING TO Graphics Plane
X
X
S45
CURRENT REPRODUCING POINT IS AT PTS IN ICS?
YES
NO
S52
INITIAL DISPLAY EXCUTION BY STARTING OUTPUT OF CONTENTS OF Graphics Plane8
SET ALL animation(p)(p=1,2.···n) AS "0"(i.e.ZERO SETTING)
S53
X
animation(p) : GLOBAL VARIABLE INDICATING ANIMATION FRAME NUMBER OF BUTTON P
S46
CURRENT REPRODUCI NG POINT IS AT selection _time_out_pts?
YES
NO
S54
AUTOMATIC ACTIVATION OF DEFAULT BUTTON
X
S47
CURRENT REPRODUCING POINT IS AT composition _time_out_pts?
YES
NO
S55
SCREAN CLEARING
X
RETURN
α = PERIOD OF WRITING EACH GRAPHICS CONSTITUTING INITIAL DISPLAY, TO GraphicsPlane+GraphicsPlane CLEARING PERIOD

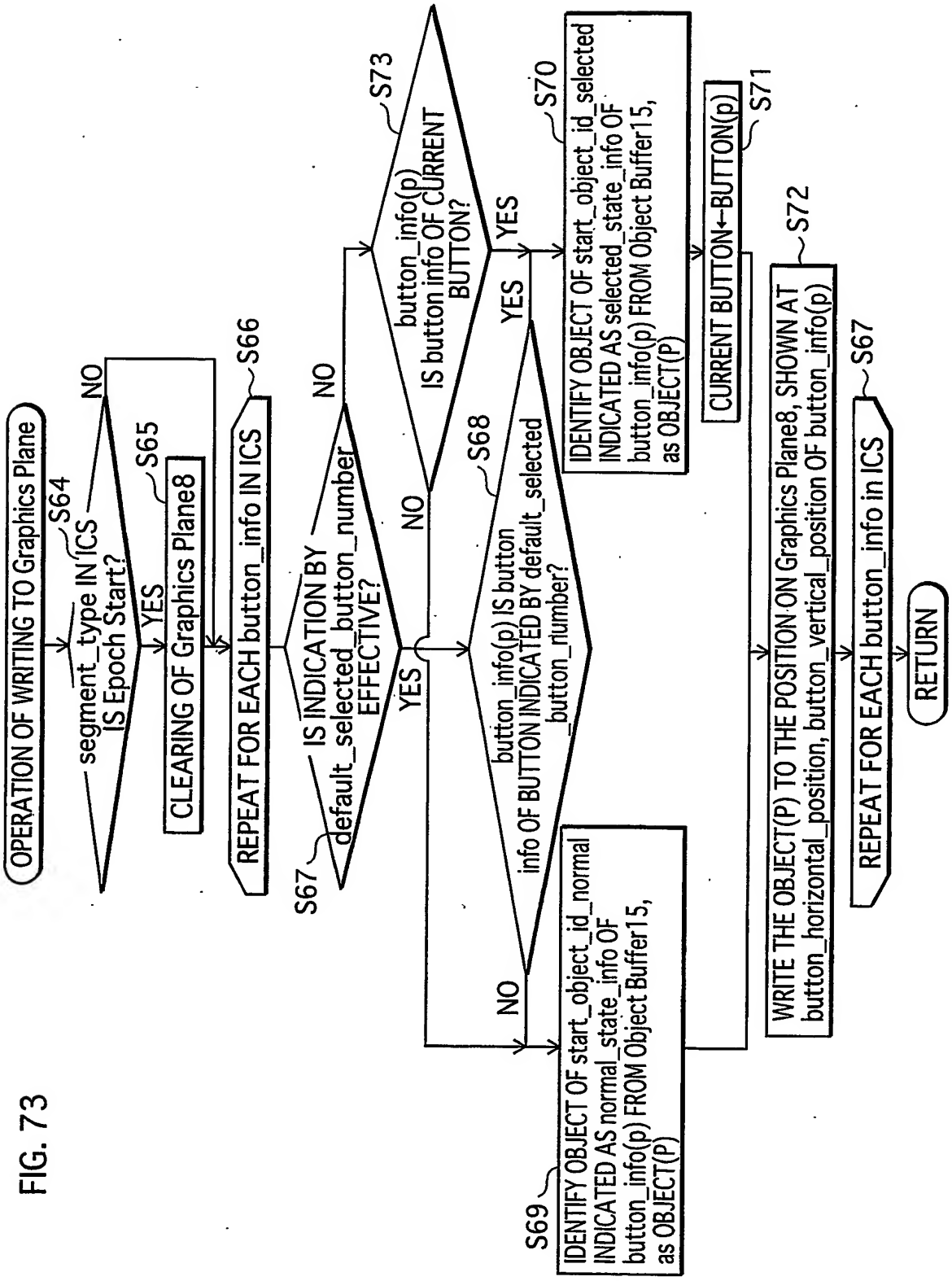

OPERATION OF WRITING TO Graphics Plane
S64
segment_type IN ICS IS Epoch Start?
NO
YES
S65
CLEARING OF Graphics Plane8
S66
REPEAT FOR EACH button_info IN ICS
S67
IS INDICATION BY default_selected_button_number EFFECTIVE?
NO
YES
S73
button_info(p) IS button info OF CURRENT BUTTON?
NO
YES
S68
button_info(p) IS button info OF BUTTON INDICATED BY default_selected button_number?
NO
YES
S69
IDENTIFY OBJECT OF start_object_id_normal INDICATED AS normal_state_info OF button_info(p) FROM Object Buffer15, as OBJECT(P)
S70
IDENTIFY OBJECT OF start_object_id_selected INDICATED AS selected_state_info OF button_info(p) FROM Object Buffer15, as OBJECT(P)
S71
CURRENT BUTTON←BUTTON(p)
S72
WRITE THE OBJECT(P) TO THE POSITION ON Graphics Plane8, SHOWN AT button_horizontal_position, button_vertical_position OF button_info(p)
S67
REPEAT FOR EACH button_info in ICS
RETURN

FIG. 73

FIG.74

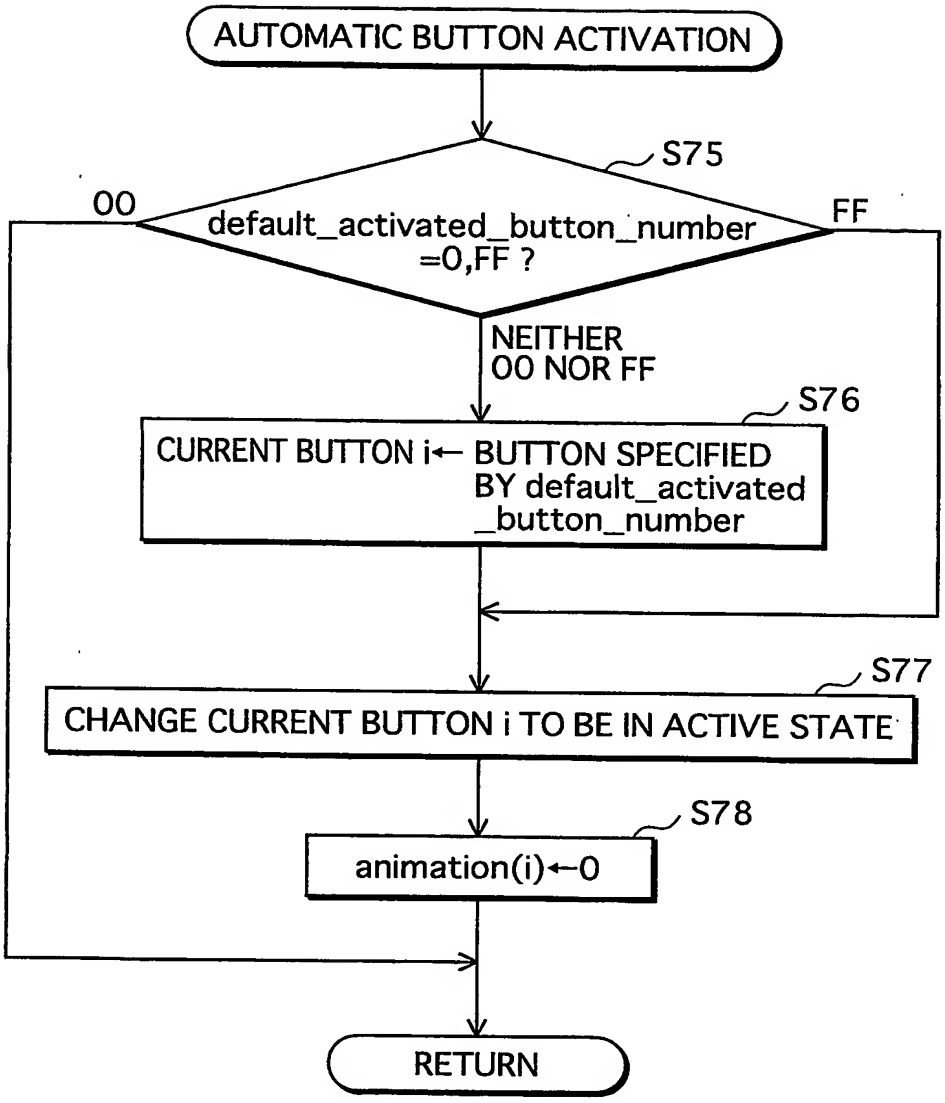
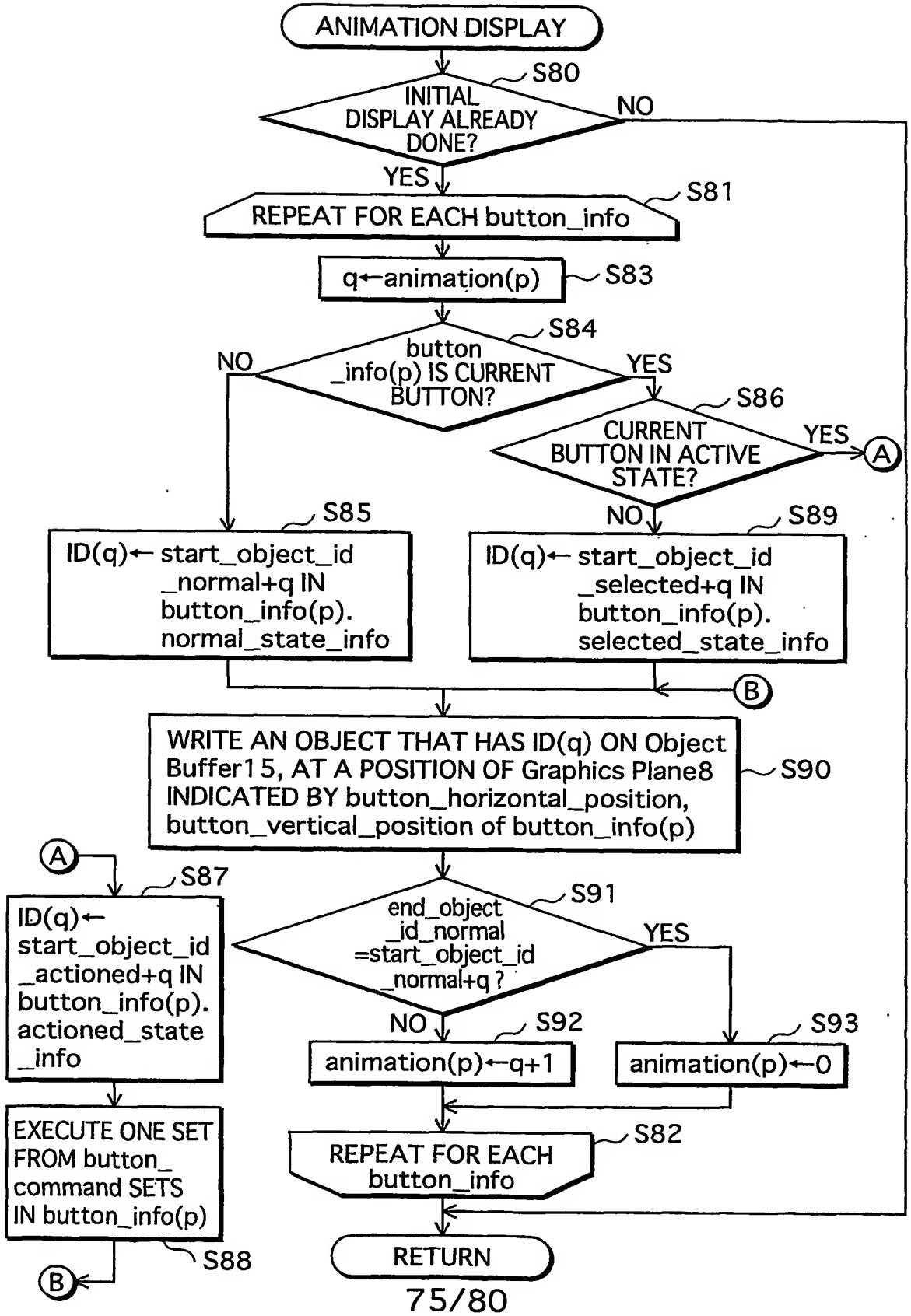

AUTOMATIC BUTTON ACTIVATION
S75
default_activated_button_number =0,FF ?
00
FF
NEITHER 00 NOR FF
S76
CURRENT BUTTON i← BUTTON SPECIFIED BY default_activated _button_number
S77
CHANGE CURRENT BUTTON i TO BE IN ACTIVE STATE
S78
animation(i)←0
RETURN

FIG.75


ANIMATION DISPLAY
S80
INITIAL DISPLAY ALREADY DONE?
NO
YES
S81
REPEAT FOR EACH button_info
q←animation(p)
S83
S84
button _info(p) IS CURRENT BUTTON?
NO
YES
S86
CURRENT BUTTON IN ACTIVE STATE?
YES
A
NO
S85
ID(q)← start_object_id _normal+q IN button_info(p). normal_state_info
S89
ID(q)← start_object_id _selected+q IN button_info(p). selected_state_info
B
WRITE AN OBJECT THAT HAS ID(q) ON Object Buffer15, AT A POSITION OF Graphics Plane8 INDICATED BY button_horizontal_position, button_vertical_position of button_info(p)
S90
A
S87
ID(q)← start_object_id _actioned+q IN button_info(p). actioned_state _info
S91
end_object _id_normal =start_object_id _normal+q ?
YES
NO
S92
animation(p)←q+1
S93
animation(p)←0
EXECUTE ONE SET FROM button_ command SETS IN button_info(p)
S88
B
S82
REPEAT FOR EACH button_info
RETURN

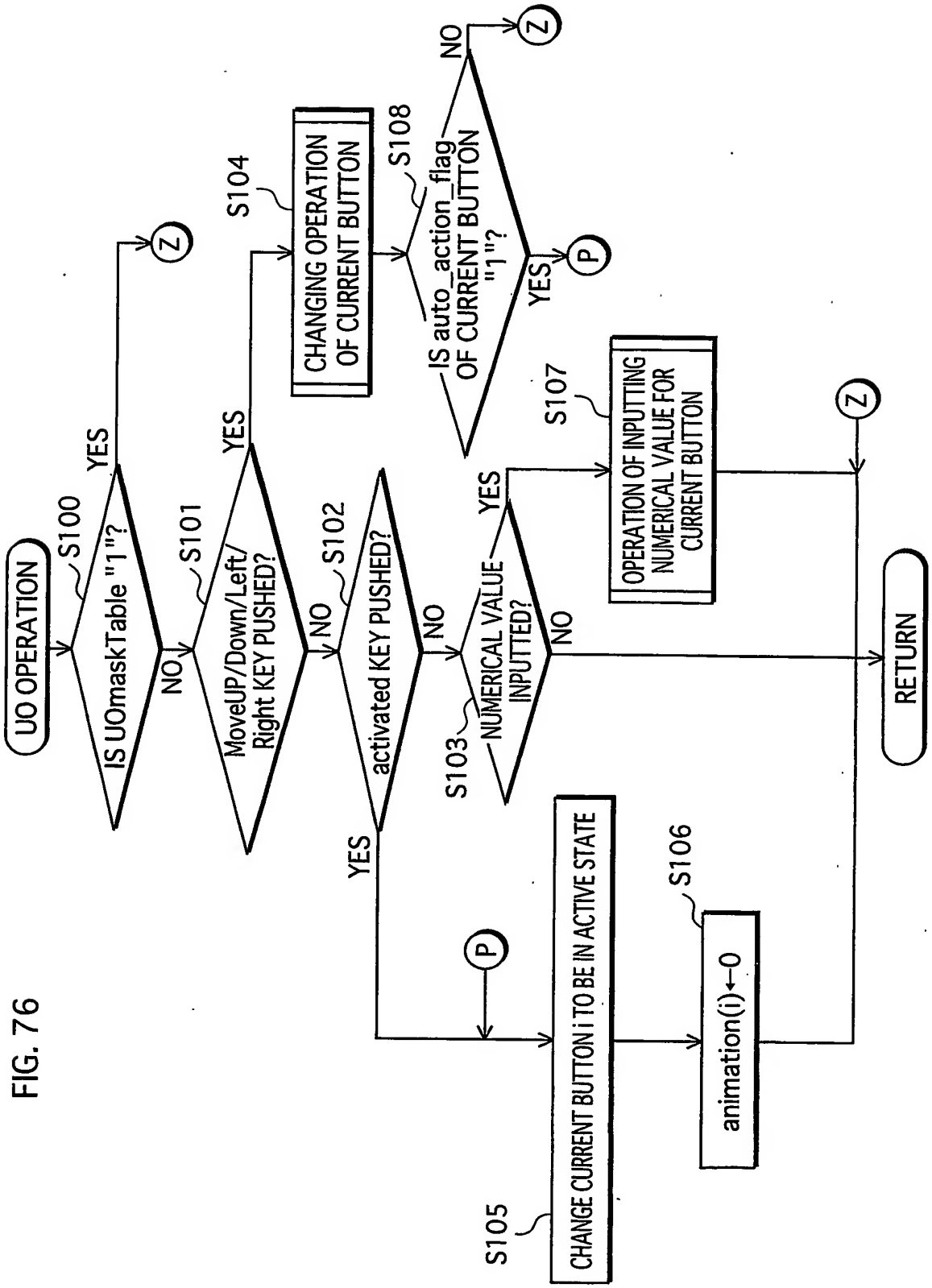
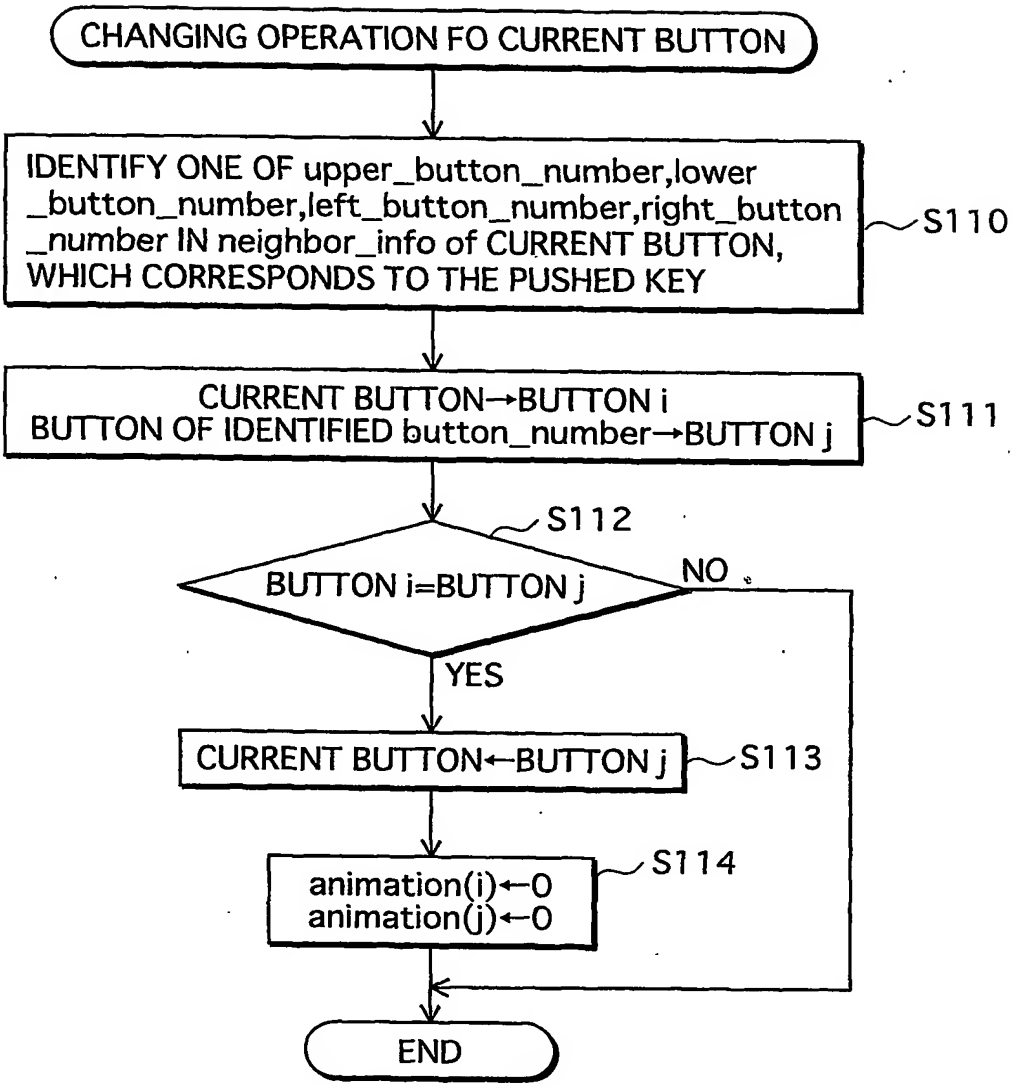

UO OPERATION
S100
IS UOmaskTable "1"?
YES
NO
Z
S101
MoveUP/Down/Left/Right KEY PUSHED?
YES
NO
S102
activated KEY PUSHED?
YES
NO
S103
NUMERICAL VALUE INPUTTED?
YES
NO
S104
CHANGING OPERATION OF CURRENT BUTTON
S108
IS auto_action_flag OF CURRENT BUTTON "1"?
YES
NO
P
Z
S107
OPERATION OF INPUTTING NUMERICAL VALUE FOR CURRENT BUTTON
Z
RETURN
P
S105
CHANGE CURRENT BUTTON i TO BE IN ACTIVE STATE
S106
animation(i)←0

FIG. 76

FIG.77


CHANGING OPERATION FO CURRENT BUTTON
IDENTIFY ONE OF upper_button_number,lower _button_number,left_button_number,right_button _number IN neighbor_info of CURRENT BUTTON, WHICH CORRESPONDS TO THE PUSHED KEY
S110
CURRENT BUTTON→BUTTON i
BUTTON OF IDENTIFIED button_number→BUTTON j
S111
S112
BUTTON i=BUTTON j
NO
YES
CURRENT BUTTON←BUTTON j
S113
animation(i)←0
animation(j)←0
S114
END

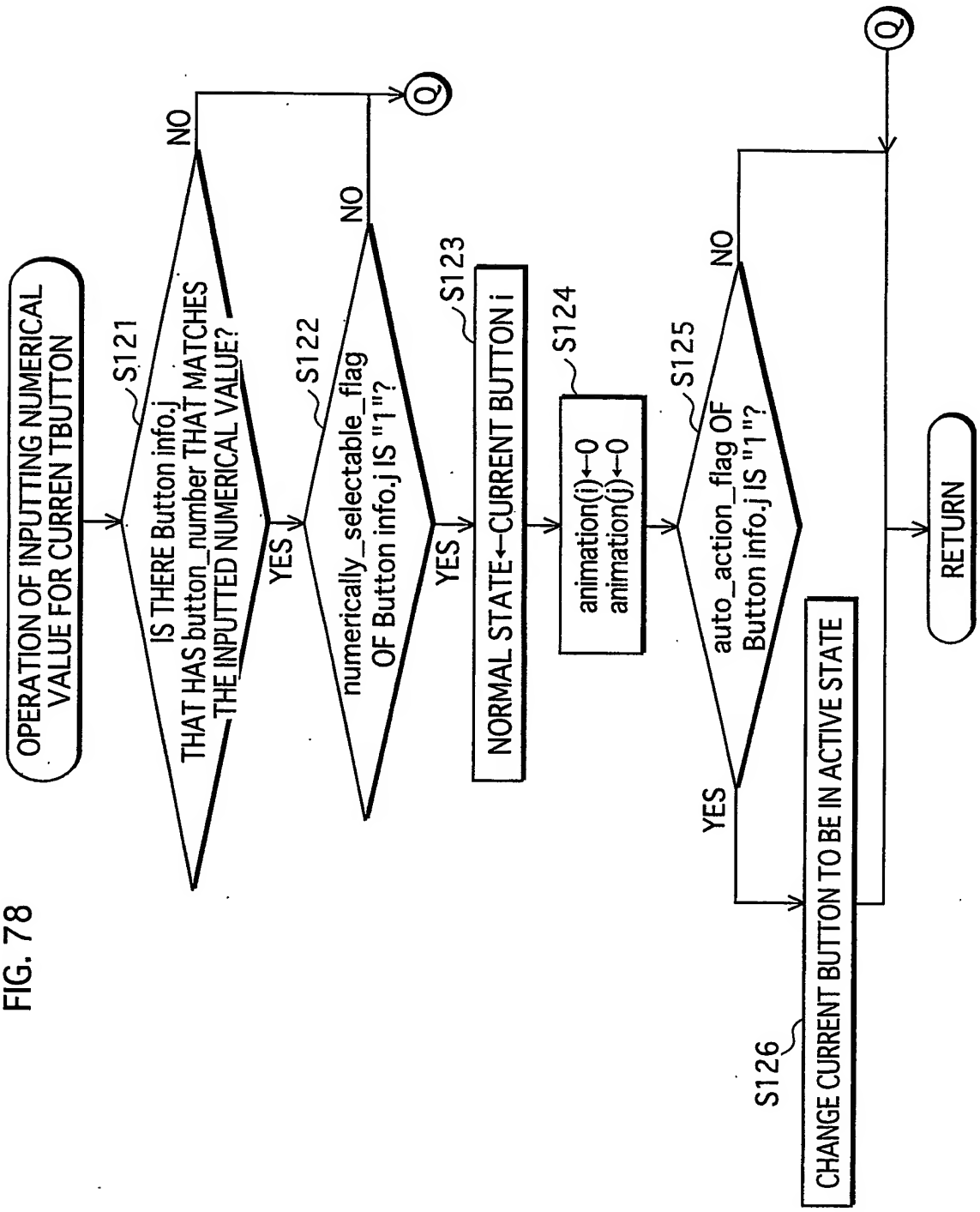

OPERATION OF INPUTTING NUMERICAL VALUE FOR CURREN TBUTTON
S121
IS THERE Button info.j THAT HAS button_number THAT MATCHES THE INPUTTED NUMERICAL VALUE?
NO
YES
S122
numerically_selectable_flag OF Button info.j IS "1"?
NO
YES
Q
S123
NORMAL STATE←CURRENT BUTTON i
S124
animation(i)←0
animation(j)←0
S125
auto_action_flag OF Button info.j IS "1"?
NO
YES
S126
CHANGE CURRENT BUTTON TO BE IN ACTIVE STATE
Q
RETURN

FIG. 78

FIG. 79

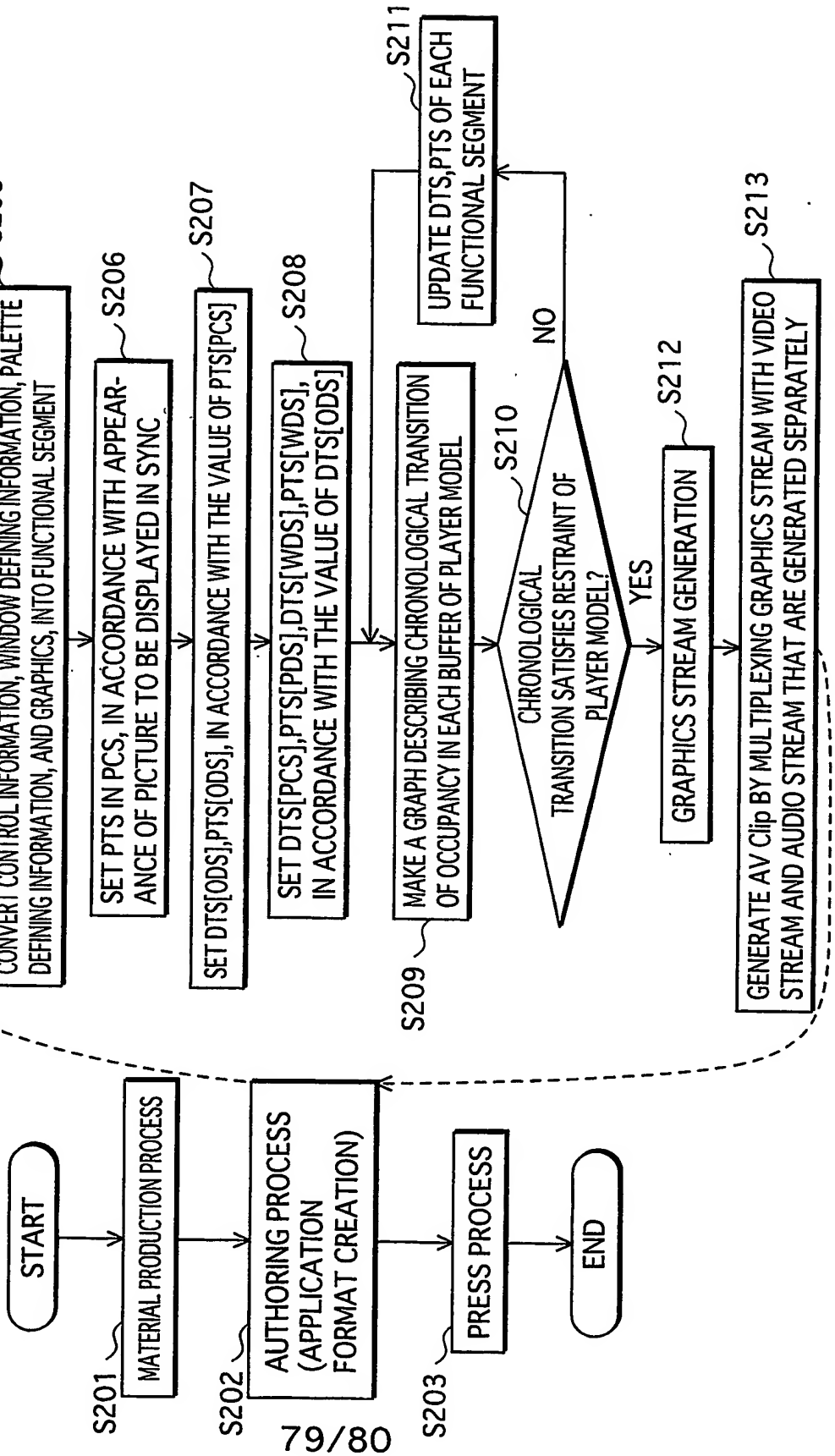
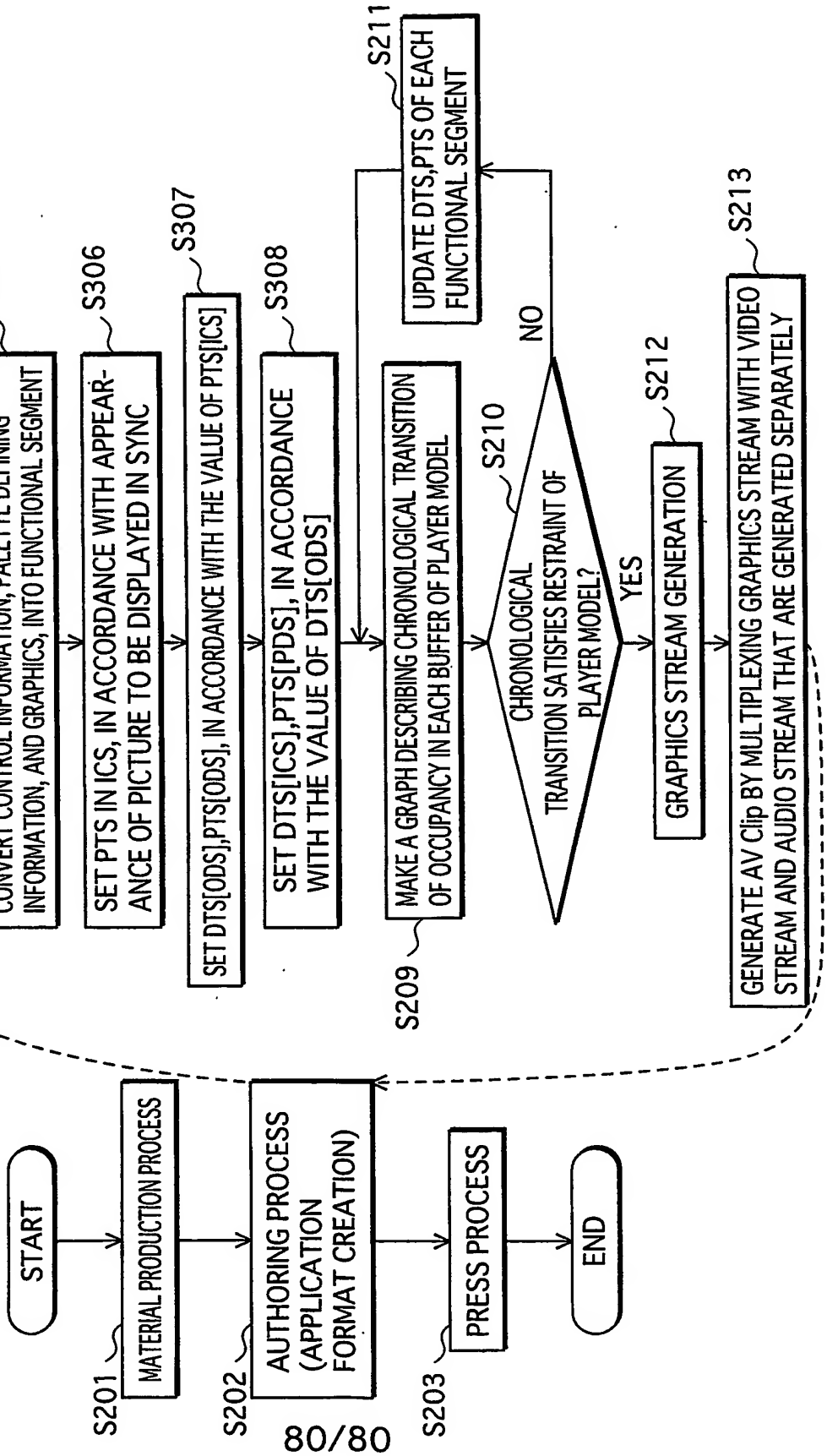

START
S201
MATERIAL PRODUCTION PROCESS
S202
AUTHORING PROCESS (APPLICATION FORMAT CREATION)
S203
PRESS PROCESS
END
S204
DESCRIBE CONTROL INFORMATION, WINDOW DEFINING INFORMATION, PALETTE DEFINING INFORMATION, AND GRAPHICS
S205
CONVERT CONTROL INFORMATION, WINDOW DEFINING INFORMATION, PALETTE DEFINING INFORMATION, AND GRAPHICS, INTO FUNCTIONAL SEGMENT
S206
SET PTS IN PCS, IN ACCORDANCE WITH APPEARANCE OF PICTURE TO BE DISPLAYED IN SYNC
S207
SET DTS[ODS],PTS[ODS], IN ACCORDANCE WITH THE VALUE OF PTS[PCS]
S208
SET DTS[PCS],PTS[PDS],DTS[WDS],PTS[WDS], IN ACCORDANCE WITH THE VALUE OF DTS[ODS]
S209
MAKE A GRAPH DESCRIBING CHRONOLOGICAL TRANSITION OF OCCUPANCY IN EACH BUFFER OF PLAYER MODEL
S210
CHRONOLOGICAL TRANSITION SATISFIES RESTRAINT OF PLAYER MODEL?
NO
YES
S211
UPDATE DTS,PTS OF EACH FUNCTIONAL SEGMENT
S212
GRAPHICS STREAM GENERATION
S213
GENERATE AV Clip BY MULTIPLEXING GRAPHICS STREAM WITH VIDEO STREAM AND AUDIO STREAM THAT ARE GENERATED SEPARATELY

FIG. 80


START
S201
MATERIAL PRODUCTION PROCESS
S202
AUTHORING PROCESS (APPLICATION FORMAT CREATION)
S203
PRESS PROCESS
END
S304
DESCRIBE CONTROL INFORMATION, PALETTE DEFINING INFORMATION, AND GRAPHICS
S305
CONVERT CONTROL INFORMATION, PALETTE DEFINING INFORMATION, AND GRAPHICS, INTO FUNCTIONAL SEGMENT
S306
SET PTS IN ICS, IN ACCORDANCE WITH APPEARANCE OF PICTURE TO BE DISPLAYED IN SYNC
S307
SET DTS[ODS],PTS[ODS], IN ACCORDANCE WITH THE VALUE OF PTS[ICS]
S308
SET DTS[ICS],PTS[PDS], IN ACCORDANCE WITH THE VALUE OF DTS[ODS]
S209
MAKE A GRAPH DESCRIBING CHRONOLOGICAL TRANSITION OF OCCUPANCY IN EACH BUFFER OF PLAYER MODEL
S210
CHRONOLOGICAL TRANSITION SATISFIES RESTRAINT OF PLAYER MODEL?
NO
YES
S211
UPDATE DTS,PTS OF EACH FUNCTIONAL SEGMENT
S212
GRAPHICS STREAM GENERATION
S213
GENERATE AV Clip BY MULTIPLEXING GRAPHICS STREAM WITH VIDEO STREAM AND AUDIO STREAM THAT ARE GENERATED SEPARATELY